

P19483.P04

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant :H. TAKAHASHI

Serial No. :Not Yet Assigned

Filed :Concurrently Herewith


For :MICROCOMPUTER WITH PROGRAM-REVISION ABILITY
CLAIM OF PRIORITY

Commissioner of Patents and Trademarks
Washington, D.C. 20231

Sir:

Applicant hereby claims the right of priority granted pursuant to 35 U.S.C. 119 based upon Japanese Application No. 11-226584, filed August 10, 1999. As required by the Statute, a certified copy of the Japanese application is being submitted herewith.

Respectfully submitted,
H. TAKAHASHI


Bruce H. Bernstein
Reg. No. 29,027

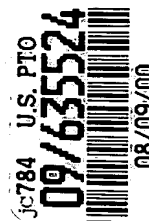
Reg No 33,328

August 9, 2000
GREENBLUM & BERNSTEIN, P.L.C.
1941 Roland Clarke Place
Reston, VA 20191
(703) 716-1191



日 本 国 特 許 庁

PATENT OFFICE
JAPANESE GOVERNMENT



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application:

1999年 8月10日

出 願 番 号

Application Number:

平成11年特許願第226584号

出 願 人

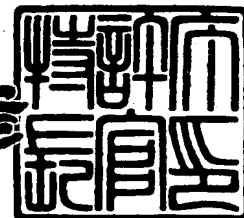
Applicant (s):

旭光学工業株式会社

2000年 4月28日

特許庁長官
Commissioner,
Patent Office

近 藤 隆 彦



出証番号 出証特2000-3032239

【書類名】 特許願

【整理番号】 AP98479

【提出日】 平成11年 8月10日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 9/06
G03B 7/00
G06F 15/04

【発明者】

【住所又は居所】 東京都板橋区前野町2丁目36番9号 旭光学工業株式会社内

【氏名】 高橋 宏之

【特許出願人】

【識別番号】 000000527

【氏名又は名称】 旭光学工業株式会社

【代表者】 松本 徹

【代理人】

【識別番号】 100090169

【弁理士】

【氏名又は名称】 松浦 孝

【手数料の表示】

【予納台帳番号】 050898

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9002979

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 プログラム修正機能内蔵マイクロコンピュータ

【特許請求の範囲】

【請求項 1】 プログラムを格納する読出し専用メモリ手段と、前記プログラムの一連のアドレスに逐次アクセスして命令を取り込み、該命令を解読して逐次処理を行う制御／演算手段と、この制御／演算手段によってアクセスすべきアドレスを逐次更新するプログラムカウンタ手段とを具備して成るマイクロコンピュータにおいて、

前記プログラムに従って前記制御／演算手段によって実行される処理に割込み処理を発生させるべき割込み発生アドレスを格納する少なくとも 1 つの割込み発生アドレス格納手段と、

前記プログラムカウンタ手段で逐次更新されるアドレスを前記割込み発生アドレス格納手段の割込み発生アドレスと比較するアドレス比較手段と、

前記割込み発生アドレス格納手段の割込み発生アドレスと対応関係にある割込み先アドレスを格納する割込み先アドレス格納手段と、

前記アドレス比較手段による双方のアドレスの一致時にその割込み発生アドレスと対応関係にある割込み先アドレスを指定して所定の割込み処理を実行させる割込み先アドレス指定手段とが設けられることを特徴とするマイクロコンピュータ。

【請求項 2】 請求項 1 に記載のマイクロコンピュータにおいて、更に、前記アドレス比較手段による双方のアドレスの一致が適正か否かを判別する判別手段と、この判別手段により前記アドレス比較手段による双方のアドレスの一致が不適正と判別されたとき前記割込み先アドレス指定手段を無効化する無効化手段とが設けられることを特徴とするマイクロコンピュータ。

【請求項 3】 請求項 1 または 2 に記載のマイクロコンピュータにおいて、更に、前記割込み発生アドレス及び前記割込み先アドレスを格納する書換え可能なメモリ手段が設けられ、マイクロコンピュータの立上げ時毎に前記書換え可能なメモリ手段から前記割込み発生アドレス及び前記割込み先アドレスが前記割込

み発生アドレス格納手段及び前記割込み先アドレス格納手段に取り込まれることを特徴とするマイクロコンピュータ。

【請求項4】 請求項1から3までのいずれか1項に記載のマイクロコンピュータにおいて、前記アドレス比較手段が前記プログラムカウンタ手段に直接的に接続されて、該プログラムカウンタ手段からアドレスを逐次取り込むことを特徴とするマイクロコンピュータ。

【請求項5】 請求項1から3までのいずれか1項に記載のマイクロコンピュータにおいて、前記アドレス比較手段が前記プログラムカウンタ手段からのデータバスに接続され、そのデータバスからアドレスを逐次取り込むことを特徴とするマイクロコンピュータ。

【請求項6】 請求項1から5までのいずれか1項に記載のマイクロコンピュータにおいて、前記割込み先アドレス指定手段が前記アドレス比較手段による双方のアドレスの一致時にその割込み発生アドレスと対応関係にある割込み先アドレスを前記割込み先アドレス格納手段から読み出して前記プログラムカウンタ手段に書き込み、これにより前記所定の割込み処理が実行されることを特徴とするマイクロコンピュータ。

【請求項7】 請求項1から5までのいずれか1項に記載のマイクロコンピュータにおいて、前記割込み先アドレス指定手段が前記アドレス比較手段による双方のアドレスの一致時にその割込み発生アドレスと対応関係にある割込み先アドレスを前記割込み先アドレス格納手段から読み出して一時的に保持するアドレス保持手段を具備して、このアドレス保持手段に保持された割込み先アドレスにアクセスすることを前記制御／演算手段に指令し、これにより前記所定の割込み処理が実行されることを特徴とするマイクロコンピュータ。

【請求項8】 請求項1から7までのいずれか1項に記載のマイクロコンピュータにおいて、前記所定の割込み処理の実行が完了された直後に前記プログラムカウンタ手段に前記割込み発生アドレスが再設定された際に前記アドレス比較手段によるアドレスの一致を一時的に無効化するアドレス一致一時無効化手段が設けられることを特徴とするマイクロコンピュータ。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、プログラム格納メモリが基板上から容易に取り外し得ないように搭載されているマイクロコンピュータ、特にマイクロコンピュータの全ての機能を一チップ上に集積化した所謂ワン・チップ・マイクロコンピュータであって、プログラム修正機能を持つマイクロコンピュータに関する。

【0 0 0 2】

【従来の技術】

周知のように、マイクロコンピュータは中央演算処理ユニット（CPU）、読出し専用メモリ（ROM）、書込み／読出し自在なメモリ（RAM）及び入出力インターフェース（I/O）等を具備し、これら構成要素は配線基板上に搭載される。一方、ワン・チップ・マイクロコンピュータでは、かかる構成要素は一チップ上に一体的に集積化されて搭載される。ROMが配線基板に容易に取り外し得ないように搭載されているマイクロコンピュータやワン・チップ・マイクロコンピュータでは、その製造工程中にROMには所定の種々のプログラムが前もって書き込まれて格納されているので、かかるマイクロコンピュータの製造後に該プログラムに対して修正や変更を加えることは多大な作業労力が必要となる。

【0 0 0 3】

従って、プログラムに所謂バグ等が発見された場合やマイクロコンピュータ及びそれを使用する電子機器製品の製造中或いは完成後にプログラムの修正や変更が生じた場合には、マイクロコンピュータ及びそれを使用する電子機器製品の製造を最初からやり直さなければならず、この点が上述したようなタイプのマイクロコンピュータに伴う製造コスト面での大きな問題点として指摘されている。勿論、ROMが取外し自在に配線基板上に搭載されるようなタイプのマイクロコンピュータの場合には、ROMを取り外してプログラム書込み装置に設置してそのプログラムに対して修正或いは変更を行うことは可能である。

【0 0 0 4】

なお、説明の便宜上、本明細書では、特に判らない限り、マイクロコンピュータという用語はROMが配線基板から容易に取り外し得ないように搭載されてい

るマイクロコンピュータ及びワン・チップ・マイクロコンピュータを指すものとする。

【0005】

上述したような問題に対処する方策として、従来では、電氣的に消去可能でしかもプログラム可能なメモリ（EEPROM）をマイクロコンピュータとを組み合わせ、EEPROMに必要な応じて修正プログラム或いは追加プログラムを格納し、マイクロコンピュータの立上げ時にEEPROMから修正プログラム或いは追加プログラムをRAMに取り込み、ROMのプログラムの実行時にRAMの修正プログラム或いは追加プログラムを実行させることが提案されている（例えば、特開平8-272601号公報）。

【0006】

詳述すると、ROMには種々のプログラムが格納されており、それらプログラムのうちの幾つか、例えばバグ等の発生し易いプログラムや或いは将来修正や変更を加え得る可能性のあるプログラムへの導入箇所に修正／追加プログラム呼出し命令が書き込まれ、CPUでの演算処理中にその修正／追加プログラム読出し命令が読み取られると、演算処理は一旦中断し、修正／追加プログラムが実行不可となっていた場合には、ROMのプログラムに直ちに戻ってその実行が継続される。一方、修正／追加プログラムが実行可能となっていれば、その修正プログラム或いは追加プログラムが実行され、その実行終了後に再びROMのプログラムの所定箇所に戻って、所定のプログラムが実行される。要するに、後者の場合には、ROMのプログラムのうちの所定のプログラム（例えば、バグ等の発生したプログラム）がRAMの所定アドレスに格納された修正プログラム或いは追加プログラムによって見かけ上置き換えられたことになる。

【0007】

以上のような構成によれば、EEPROMに修正プログラム或いは追加プログラムを必要に応じて書き込むことにより、ROMの所定のプログラムに対して修正或いは変更を加えることが可能であり、かくしてマイクロコンピュータの製造後にROMのプログラムにバグ等が発見された場合でも或いは所定のプログラムに修正や変更があった場合でも、マイクロコンピュータの製造及びその電子機器

の製造を最初からやり直さなければならないというような面倒な問題は回避され得る。

【0008】

【発明が解決しようとする課題】

しかしながら、従来の場合にあっては、ROMのプログラムの修正或いは変更を行うためには、その対象となるプログラムへの導入箇所に予め修正／追加プログラム呼出し命令を書き込んでおくことが必要であり、修正／追加プログラム呼出し命令の書込みの無いプログラムに対する修正或いは変更は行い得ない。要するに、従来の対処方法では、後で修正が必要となるであろうと予想される所定のプログラムの幾つかについてのみ修正或いは変更が可能であるに過ぎず、修正箇所に対する自由度が高くない。

【0009】

勿論、ROMの全てのプログラムの導入箇所に修正／追加プログラム呼出し命令を書き込むことにより、全てのプログラムに対して修正或いは変更を加えることは可能となるが、その場合にはプログラムのあらゆる導入箇所に修正／追加プログラム呼出し命令を書き込まなければならず、マイクロコンピュータの処理時間が徒に増大させられることになるので現実的ではない。

【0010】

従って、本発明の目的は、プログラム修正機能を内蔵したマイクロコンピュータであって、その処理時間に実質的な影響を与えることなく、読出し専用メモリ（ROM）の任意のプログラムに対して修正或いは変更を必要に応じて加え得るように構成されたマイクロコンピュータを提供することである。

【0011】

【課題を解決するための手段】

本発明によるマイクロコンピュータはプログラムを格納する読出し専用メモリ手段と、該プログラムの一連のアドレスに逐次アクセスして命令を取り込み、その命令を解読して逐次処理を行う制御／演算手段と、この制御／演算手段によってアクセスすべきアドレスを逐次更新するプログラムカウンタ手段とを具備して成るものである。本発明によれば、このようなマイクロコンピュータにおいて、

制御／演算手段によって実行される処理に割込み処理を発生させるべき少なくとも1つの割込み発生アドレスを格納する割込み発生アドレス格納手段と、このプログラムカウンタ手段で逐次更新されるアドレスを割込み発生アドレス格納手段の割込み発生アドレスと比較するアドレス比較手段と、割込み発生格納手段の割込み発生アドレスと対応関係にある割込み先アドレスを格納する割込み先アドレス格納手段と、アドレス比較手段による双方のアドレスの一致時にその割込み発生アドレスと対応関係にある割込み先アドレスを指定して所定の割込み処理を実行させる割込み先アドレス指定手段とを設けることが特徴とされる。

【0012】

本発明による好適な実施形態においては、アドレス比較手段による双方のアドレスの一致が適正か否かを判別する判別手段と、この判別手段によりアドレス比較手段での双方のアドレスの一致が不適正と判別されたとき割込み先アドレス指定手段を無効化する無効化手段とが設けられる。

【0013】

また、本発明による好適な実施形態においては、割込み発生アドレス及び割込み先アドレスを格納する書換え可能なメモリ手段が設けられ、マイクロコンピュータの立上げ時毎に書換え可能なメモリ手段から割込み発生アドレス及び割込み先アドレスが割込み発生アドレス格納手段及び割込み先アドレス格納手段に取り込まれる。

【0014】

本発明において、アドレス比較手段はプログラムカウンタ手段に直接的に接続されてそこからアドレスを逐次取り込むこともできるし、またプログラムカウンタ手段からのデータバスに接続されてそのデータバスからアドレスを逐次取り込んでもよい。

【0015】

本発明の好ましい実施形態においては、割込み先アドレス指定手段がアドレス比較手段での双方のアドレスの一致時にその割込み発生アドレスと対応関係にある割込み先アドレスを割込み先アドレス格納手段から読み出してプログラムカウンタに書き込み、これにより所定の割込み処理が実行されてもよく、或いは別の

好ましい実施形態では、割込み先アドレス指定手段はアドレス比較手段での双方のアドレスの一致時にその割込み発生アドレスと対応関係にある割込み先アドレスを割込み先アドレス格納手段から読み出して一時的に保持するアドレス保持手段を具備して、このアドレス保持手段に保持された割込み先アドレスにアクセスすることを制御／演算手段に指令し、これにより所定の割込み処理が実行されてもよい。

【0016】

また、本発明の好ましい実施形態においては、所定の割込み処理の実行が完了された直後にプログラムカウンタ手段に割込み発生アドレスが再設定された際にアドレス比較手段によるアドレスの一致を一時的に無効化するアドレス一致一時無効化手段が設けられ、これにより該所定の割込み処理の実行が再度実行されないようにされる。

【0017】

【発明の実施の形態】

次に、添付図面を参照して、本発明によるプログラム修正機能内蔵マイクロコンピュータの実施形態について説明する。

【0018】

図1を参照すると、本発明によるマイクロコンピュータのブロック図が示され、同図に示すように、マイクロコンピュータは中央演算処理ユニット（CPU）10、種々のルーチンを実行するためのプログラム、常数等を格納する読出し専用メモリ（ROM）12、データや修正プログラム或いは追加プログラム等を一時的に格納する書込み／読出し自在なメモリ（RAM）14、入出力インターフェース（I/O）16から成る。I/O16には周辺機器等（図示されない）が接続され、これにより周辺機器とCPU10との間で種々の信号やデータ等の送受信が行われる。本発明によれば、I/O16には電氣的に消去可能でしかもプログラム可能な読出し専用メモリ（EEPROM）18が接続され、このEEPROM18には後述するように修正プログラム或いは追加プログラム及びそれらに関連したフラグデータ等が必要に応じて書き込まれる。

【0019】

図1に示すように、CPU10には、その主要な構成要素の一つとして、制御ユニット(CU)20が設けられ、この制御ユニット20はクロック発生器22を備え、このクロック発生器22からは所定周波数の基本クロック信号が出力され、制御ユニット20は基本クロック信号を適宜分周して必要な制御クロックパルスを発生させる。また、制御ユニット20はリセット回路24を備え、CPU10の立上げ時、即ちマイクロコンピュータの電源オン時、リセット回路24から制御ユニット20にリセット信号が出力され、これによりCPU10内の各部で必要な初期化が行われる。

【0020】

制御ユニット20はプログラムカウンタ26、命令レジスタ28及び命令デコーダ30と協働してROM12内のプログラムの命令を逐次解読して、その命令に基づいてCPU10内の各部に必要な制御信号を発生して送信する。詳述すると、ROM12内には一連の命令がプログラム化されてそれぞれ所定アドレスに格納されており、制御ユニット20はプログラムカウンタ26で示された所定アドレスの命令をROM12から読み出して命令レジスタ28に保持し、次いでその命令を命令デコーダ30で解読し、その解読命令に基づいて制御信号を発生して送信する。ROM12の所定アドレスの命令が読み出されてその命令が実行される度に、プログラムカウンタ26の内容に例えば数値1が加えられ、これによりROM12から次に読み出されるべき命令のアドレスが更新される。なお、本実施形態では、ROM12内の命令のアドレスは16ビット構成とされる。

【0021】

また、CPU10には、その主要な構成要素の一つとして、算術論理演算ユニット(ALU)32が設けられ、このALU32では、2進データの演算が制御ユニット20からの制御信号に基づいて行われる。演算されるべき2進データはテンポラリレジスタ34、アキュムレータ36及び状態レジスタ38からALU32に取り込まれる。テンポラリレジスタ34に保持されるデータとしては、例えばROM12からの定数データ或いはRAM14からの一時的なデータ等が挙げられる。アキュムレータ36はそれ自身で演算機能を備え、そこにデータが入力される度にそのデータと前のデータとの演算結果を保持し、必要に応じてその

演算結果はALU32に取り込まれる。

【0022】

状態レジスタ38はALU32の演算時の種々のステータス情報をフラグデータとして保持するものであり、そこには演算結果の状態を示す種々のフラグ（例えば、例えばキャリーフラグ、オーバーフローフラグ、ゼロフラグ、サインフラグ及びサインフラグ等）が割り当てられる。また、状態レジスタ38には演算結果の状態を示す種々のフラグの他に割込み許可フラグも割り当てられ、この割込み許可フラグはリセット後に“0”とされているが、プログラム上必要に応じて割込み許可フラグに“1”が書き込まれ、このとき種々の割込み処理が許可されるが、しかし割込み許可フラグが“1”から“0”に書き換えられると、緊急時の割込み処理（NMI(non-maskable interruption)）及びソフトウェア割込み処理以外の全割込み処理が禁止される。

【0023】

図1に示すように、CPU10には更に割込み制御部40、汎用レジスタ42及びスタックポインタ44も設けられる。割込み制御部40は制御ユニット20からの割込み信号或いはI/O16に接続された周辺機器（図示されない）からの割込み信号によって割込み処理やサブルーチンの実行を制御するものである。また、汎用レジスタ42は多目的レジスタとしての機能を有し、例えば汎用レジスタ42の一部はアキュムレータ、インデックスレジスタやスタックメモリとして利用される。スタックポインタ44はスタックメモリのどこからデータを取り出すかを指示するレジスタであり、例えばサブルーチンや割込み処理が行われた後にプログラムの戻りアドレスがスタックメモリに格納される。なお、マルチプレクサ46はプログラムカウンタ26、汎用レジスタ42及びスタックポインタ44へのデータや制御信号の入力を振り分けるためのに使用される。

【0024】

以上で述べたCPU10の種々の構成要素自体は従来のマイクロコンピュータの構成要素として周知のものであり、本発明によれば、CPU10にアドレス一致割込み制御部48を新たに加える点が新規とされ、このアドレス一致割込み制御部48の種々の実施形態について以下に説明する。

【 0 0 2 5 】

アドレス一致割込み制御部 4 8 の第 1 の実施形態にあつては、アドレス一致割込み制御部 4 8 には図 2 に示すような回路素子が 8 つ設けられ、それら 8 個の回路素子が参照符号 48_n ($n = 0, 1, \dots, 6, 7$) で代表的に示されている。なお、ROM 1 2 内のプログラムに修正を施し得る箇所は回路素子 48_n の個数に一致するので、本実施形態では、ROM 1 2 のプログラムの任意の 8 箇所に修正を施すことが可能となる。

【 0 0 2 6 】

図 2 に示すように、各回路素子 48_n には比較アドレスレジスタ 5 0 が設けられ、この比較アドレスレジスタ 5 0 は CPU 1 0 の立上げ時にリセット信号によって初期化され、また制御バス、アドレスバス及びデータバスを介してマイクロコンピュータの各部に適宜接続される。比較アドレスレジスタ 5 0 は 16 ビット構成のレジスタとされ、そのビット数は ROM 1 2 のアドレスのビット数と同じである。8 つの回路素子 48_n のそれぞれの比較アドレスレジスタ 5 0 には必要に応じて所定のアドレスが書き込まれ、この所定のアドレスとは ROM 1 2 のプログラムの修正箇所を含む一連の命令群の先頭アドレス或いは ROM 1 2 のプログラムに別のプログラムを部分的に追加すべき箇所に対応したアドレスである。なお、比較アドレスレジスタ 5 0 への所定のアドレスの書込みについては後で詳しく説明する。

【 0 0 2 7 】

比較アドレスレジスタ 5 0 はバスを介してアドレス比較器 5 2 に接続され、またプログラムカウンタ 2 6 もバスを介してアドレス比較器 5 2 に接続される。上述したように、プログラムカウンタ 2 6 では、ROM 1 2 のプログラム実行中、アドレスが逐次更新され、そのアドレスが各回路素子 48_n のアドレス比較器 5 2 で比較アドレスレジスタ 5 0 内のアドレスと比較される。アドレス比較器 5 2 の出力端子は AND ゲート 5 4 の一方の入力端子に接続され、プログラムカウンタ 2 6 で逐次更新されるアドレスが比較アドレスレジスタ 5 0 内のアドレスと一致したとき、アドレス比較器 5 2 の出力端子の出力信号はローレベルからハイレベルに変化する。即ち、プログラムカウンタ 2 6 での逐次更新アドレスが比較ア

ドレスレジスタ 5 0 内のアドレスと一致したとき、アドレス比較器 5 2 からはアドレス一致信号がハイレベル信号として出力される。なお、アドレス比較器 5 2 も CPU 1 0 の立上げ時にリセット信号によって初期化される。

【 0 0 2 8 】

図 2 に示すように、AND ゲート 5 4 の他方の入力端子はインバータ 5 6 を介してアドレス一致割込み一時禁止レジスタ 5 8 の出力端子に接続される。アドレス一致割込み一時禁止レジスタ 5 8 は 1 ビット構成のレジスタであり、制御バス、アドレスバス及びデータバスを介してマイクロコンピュータの各部に適宜接続される。アドレス一致割込み一時禁止レジスタ 5 8 には通常は 1 ビットデータとして“0”が書き込まれており、このときアドレス一致割込み一時禁止レジスタ 5 8 の出力端子からの出力信号はローレベルに維持されている。しかし、アドレス一致割込み一時禁止レジスタ 5 8 の 1 ビットデータが“0”から“1”に書き換えられると、アドレス一致割込み一時禁止レジスタ 5 8 の出力端子からの出力信号はローレベルからハイレベルに変化させられる。要するに、アドレス一致割込み一時禁止レジスタ 5 8 の 1 ビットデータは通常は“0”とされ、このときインバータ 5 6 の出力端子からはハイレベル信号が出力され、アドレス一致割込み一時禁止レジスタ 5 8 の 1 ビットデータが“1”に書き換えられると、インバータ 5 6 の出力端子からはローレベル信号が出力される。なお、アドレス一致割込み一時禁止レジスタ 5 8 の 1 ビットデータの“0”から“1”への書換えの意味については後で詳しく説明することにする。

【 0 0 2 9 】

図示するように、アドレス一致割込み一時禁止レジスタ 5 8 にはリセット回路 6 0 が組み合わされ、このリセット回路 6 0 はアドレス一致割込み一時禁止レジスタ 5 8 の 1 ビットデータが“0”から“1”に書き換えられた後にその 1 ビットデータを“1”から“0”に戻す機能を持つ。即ち、リセット回路 6 0 からアドレス一致割込み一時禁止レジスタ 5 8 にリセット信号 RE 1 がハイレベル信号として出力されたとき、アドレス一致割込み一時禁止レジスタ 5 8 の 1 ビットデータは“1”から“0”に書き戻される。リセット回路 6 0 にはアドレス比較器 5 2 からアドレス一致信号が入力されると共に CPU 1 0 の処理動作タイミング

を司る所定周波数のシステムクロックパルスが入力され、リセット回路60ではアドレス一致信号とシステムクロックパルスとに基づいてリセット信号RE1が後述するような態様で生成されてアドレス一致割込み一時禁止レジスタ58に対して出力される。なお、リセット回路60はCPU10の立上げ時にリセット信号によって初期化される。

【0030】

図2に示すように、ANDゲート54の出力端子はフリップフロップ62の入力端子Cに接続され、一方フリップフロップ62の入力端子Dにはハイレベル信号がCPU10の立上げ後に常時入力される。インバータ56からの出力信号がハイレベルであるときにアドレス比較器52からアドレス一致信号が出力されると、ANDゲート54の出力端子からの出力信号はハイレベルとなり、このときフリップフロップ62の出力端子Qからはアドレス一致割込み要求信号がハイレベル信号として出力される。

【0031】

フリップフロップ62の出力端子Qはアドレス一致割込み監視レジスタ64の入力端子に接続され、このアドレス一致割込み監視レジスタ64は1ビット構成のレジスタである。アドレス一致割込み監視レジスタ64はCPU10の立上げ時にリセット信号により初期化され、また制御バス、アドレスバス及びデータバスを介してマイクロコンピュータの各部に適宜接続される。アドレス一致割込み監視レジスタ64の1ビットデータは初期化により“0”とされているが、フリップフロップ62の出力端子Qからアドレス一致割込み要求信号（ハイレベル）が出力されると、アドレス一致割込み監視レジスタ64の1ビットデータは“0”から“1”に書き換えられ、アドレス一致割込み要求信号の出力停止（即ち、出力端子Qの出力信号がローレベルとなると）により、該1ビットデータは“1”から“0”に戻る。要するに、アドレス一致割込み監視レジスタ64の1ビットデータを読み出すことにより、フリップフロップ62の出力端子Qからのアドレス一致割込み要求信号の出力の有無を確認することができる。

【0032】

図2に示すように、フリップフロップ62の出力端子QはANDゲート66の

一方の入力端子に接続され、その他方の入力端子はアドレス一致割込み許可レジスタ 68 の出力端子に接続される。アドレス一致割込み許可レジスタ 68 は 1 ビット構成のレジスタであり、制御バス、アドレスバス及びデータバスを介してマイクロコンピュータの各部に適宜接続される。アドレス一致割込み許可レジスタ 68 の 1 ビットデータが“1”であるときには、アドレス一致割込み許可レジスタ 68 の出力端子からの出力信号はハイレベルとされる。一方、アドレス一致割込み許可レジスタ 68 の 1 ビットデータが“0”であるときには、アドレス一致割込み許可レジスタ 68 の出力端子からの出力信号はローレベルとされ、このハイレベル信号は AND ゲート 66 の他方の入力端子に出力される。要するに、フリップフロップ 62 の出力端子 Q からアドレス一致割込み要求信号が出力されたとき、アドレス一致割込み許可レジスタ 68 の 1 ビットデータが“1”とされていれば、AND ゲート 66 の入力端子の双方にはハイレベル信号が入力され、これにより AND ゲート 66 の出力端子からはアドレス一致割込み信号がハイレベル信号として割込み制御部 40 に対して出力される。割込み制御部 40 にアドレス一致割込み信号が入力されると、CPU 10 では後で詳しく説明するような割込み処理が実行される。なお、アドレス一致割込み許可レジスタ 68 は CPU 10 の立上げ時にリセット信号により初期化される。

【0033】

図 2 に示すように、AND ゲート 66 の出力端子からのアドレス一致割込み信号はベクタ・アドレス・レジスタ 70 にも入力される。ベクタ・アドレス・レジスタ 70 は CPU 10 の立上げ時にリセット信号によって初期化され、また制御バス、アドレスバス及びデータバスを介してマイクロコンピュータの各部に適宜接続される。ベクタ・アドレス・レジスタ 70 は 16 ビット構成のレジスタとされ、そのビット数は ROM 12 のアドレスのビット数と同じである。8 つの回路素子 48_n のそれぞれのベクタ・アドレス・レジスタ 70 には必要に応じて所定のベクタアドレスが書き込まれ、このベクタアドレスとは ROM 12 のプログラムの一部を修正すべき修正プログラム或いは ROM 12 のプログラムに別のプログラムを部分的に追加すべき追加プログラムをコールするための命令を格納したアドレスである。AND ゲート 66 の出力端子からのアドレス一致割込み信号がベ

クタ・アドレス・レジスタ 70 に入力されると、そのベクタアドレスはバスを介してプログラムカウンタに対して出力されてそこに書き込まれる。なお、ベクタ・アドレス・レジスタ 70 に対するベクタアドレスの書込みについては後で詳しく説明する。

【0034】

フリップフロップ 62 の出力端子 Q からアドレス一致割込み要求信号が出力された後、リセット回路 72 からはリセット信号 RE 2 がハイレベル信号としてフリップフロップ 62 のリセット端子 R に出力され、これによりフリップフロップ 62 のラッチが解除されて、出力端子 Q からのアドレス一致割込み要求信号の出力が停止される。即ち、リセット回路 72 からのリセット信号 RE 2 の出力により、フリップフロップの出力端子 Q からの出力信号がハイレベルからローレベルに戻される。リセット回路 72 にはアドレス一致割込み要求クリアレジスタ 74 からクリア信号 CL が入力されると共に CPU 10 の処理動作タイミングを司る所定周波数のシステムクロックパルスが入力され、リセット回路 72 ではクリア信号 CL とシステムクロックパルスとに基づいてリセット信号 RE 2 が後で詳述するような態様で生成されてフリップフロップ 62 のリセット端子 R に対して出力される。なお、リセット回路 72 は CPU 10 の立上げ時にリセット信号によって初期化される。

【0035】

アドレス一致割込み要求クリアレジスタ 74 は 1 ビット構成のレジスタであり、制御バス、アドレスバス及びデータバスを介してマイクロコンピュータの各部に適宜接続される。アドレス一致割込み要求クリアレジスタ 74 の 1 ビットデータは通常は“0”とされているが、AND ゲート 66 からのアドレス一致割込み信号が割込み制御部 40 に入力された後、アドレス一致割込み要求クリアレジスタ 74 の 1 ビットデータが“0”から“1”に書き換えられ、このときクリア信号がアドレス一致割込み要求クリアレジスタ 74 からハイレベル信号としてリセット回路 72 に対して出力される。なお、アドレス一致割込み要求クリアレジスタ 74 への 1 ビットデータ“1”の書込みについては後で詳しく説明する。

【0036】

上述したように、フリップフロップ62のリセット端子Rに対してリセット回路72からリセット信号RE2が出力されてフリップフロップ62のラッチが解除されると、リセット回路72からリセット信号RE3がアドレス一致割込み要求クリアレジスタ74に対して出力され、これによりアドレス一致割込み要求クリアレジスタ74の1ビットデータが“1”から“0”に書き戻され、これによりアドレス一致割込み要求クリアレジスタ74からのクリア信号の出力は停止される。即ち、アドレス一致割込み要求クリアレジスタ74からリセット回路72への出力信号がハイレベルからローレベルに変化する。なお、リセット信号RE3もリセット信号RE2と同様にクリア信号CLとシステムクロックパルスとに基づいて生成されるものであり、これについては後で詳しく説明する。

【0037】

CPU10の立上げ時、即ちCPU10の電源オン時には、比較アドレスレジスタ50、アドレス一致割込み許可レジスタ68及びベクタ・アドレス・レジスタ70のそれぞれにはEEPROM18から読み出された所定のデータが格納されて保持され、このとき同時にRAM14にはEEPROM18から読み出された修正プログラムや追加プログラム等が格納される。

【0038】

EEPROM18に対する修正プログラムや追加プログラム及びデータ等の書込みについては、EEPROM18が図1に示すようなマイクロコンピュータに対してどのように搭載されているかで異なる。例えば、EEPROM18がマイクロコンピュータの基板に恒久的に固定された状態で搭載されているとき、EEPROM18へのプログラム等の書込みについては、図1のマイクロコンピュータをEEPROM書込み用コンピュータに接続させて該マイクロコンピュータのCPU10を介して行われ、この場合にはEEPROM書込みプログラムをROM12に予め用意しておくことが必要である。一方、EEPROM18がマイクロコンピュータの基板に対して着脱自在に搭載されているとき、EEPROM18は一旦取り外され、次いでそのEEPROM18にEEPROM書込み用コンピュータでプログラム及びデータ等が直接書き込まれ、その後書込み済みEEPROM18がマイクロコンピュータの基板に搭載されることになる。勿論、後者

の場合には、ROM 1 2 に E E P R O M 書込みプログラムを ROM 1 2 に予め用意しておく必要はない。

【 0 0 3 9 】

図 3 を参照すると、E E P R O M 1 8 のメモリ領域が概念図として模式的に図示され、本実施形態においては、E E P R O M 1 8 には、8 つの 16 ビット構成の比較アドレスデータ CA0 [CA0₁₅, CA0₁₄, ...CA0₀₁, CA0₀₀]、CA1 [CA1₁₅, CA1₁₄, ...CA1₀₁, CA1₀₀]、...CA6 [CA6₁₅, CA6₁₄, ...CA6₀₁, CA6₀₀] 及び CA7 [CA7₁₅, CA7₁₄, ...CA7₀₁, CA7₀₀] を書き込むための比較アドレス書込み領域 CAZ0、CAZ1、...CAZ6 及び CAZ7 と、8 つの 16 ビット構成のベクタ・アドレス・データ VA0 [VA0₁₅, VA0₁₄, ...VA0₀₁, VA0₀₀]、VA1 [VA1₁₅, VA1₁₄, ...VA1₀₁, VA1₀₀]、...VA6 [VA6₁₅, VA6₁₄, ...VA6₀₁, VA6₀₀] 及び VA7 [VA7₁₅, VA7₁₄, ...VA7₀₁, VA7₀₀] を書き込むためのベクタアドレス書込み領域 VAZ0、VAZ1、...VAZ6 及び VAZ7 と、8 ビット分のフラグデータ FE0、FE1、...FE6 及び FE7 を書き込むためのフラグデータ書込み領域とが用意される。また、E E P R O M 1 8 には、更に、ROM 1 2 内に格納されたプログラムの一部を修正したりあるいは該プログラムについて新たなプログラムを追加するような場合の修正プログラムや追加プログラムを書き込むための修正/追加プログラム書込み領域と、図 1 に示すマイクロコンピュータを搭載すべき電子機器例えばプリンタ、ビデオカメラ、カメラや時計等に係わる種々のデータ（例えば補正データ、調整データ等）を書き込むためのデータ書込み領域とが用意される。なお、後述するように、比較アドレスデータ CA0、CA1、...CA6 及び CA7 はベクタ・アドレス・データ VA0、VA1、...VA6 及び VA7 のそれぞれと対応関係にあり、更には 8 ビット分のフラグデータ FE0、FE1、...FE6 及び FE7 のそれぞれとも対応関係にある。

【 0 0 4 0 】

本発明によれば、ROM 1 2 に格納されたプログラムの第 1 のプログラム修正例として、その中の或る一連の命令群にバグが発見された場合には、その一連の命令群によって実行されるべき正常なプログラムが E E P R O M 1 8 の修正/追加プログラム書込み領域に修正プログラムとして書き込まれる。この場合、ROM 1 2 内の一連の命令群（バグが含まれる）の実行を開始させるために必要な最

先のアドレスがEEPROM 18の比較アドレス書込み領域(CAZ0、CAZ1、…CAZ6、CAZ7)のいずれかに比較アドレスデータとして書き込まれ、また上述の修正プログラムの先頭アドレス(即ち、その修正プログラムの命令群をRAM 14に格納した際の第1番目の命令が格納されたアドレス)が該比較アドレスデータに対応したベクタ・アドレス・データとして所定のベクタデータ書込み領域(VAZ0、VAZ1、…VAZ6、VAZ7)に書き込まれる。更に、それら比較アドレスデータ及びベクタ・アドレス・データと対応関係にあるフラグデータ(FE0、FE1、…FE6、FE7)に“1”が与えられる。例えば、ROM 12内の一連の命令群(バグが含まれる)の最先アドレスが比較アドレスデータCA0として比較アドレス書込み領域CAZ0に書き込まれると、その比較アドレス書込み領域CAZ0と対応関係にあるベクタアドレス書込み領域VAZ0には上述の修正プログラムの先頭アドレスがベクタ・アドレス・データVA0として書き込まれ、その双方のアドレス書込み領域CAZ0及びVAZ0と対応関係にあるフラグデータFE0には“1”が与えられる。

【0041】

また、ROM 12内のプログラムの第2のプログラム修正例として、そのプログラムに新たなプログラムを追加することが必要とされた場合には、その新たなプログラムはEEPROM 18の修正/追加プログラム書込み領域に追加プログラムとして書き込まれ、このときROM 12内のプログラムに追加プログラムを追加すべき箇所に対応したアドレスがEEPROM 18の比較アドレス書込み領域(CAZ0、CAZ1、…CAZ6、CAZ7)のいずれかに比較アドレスデータとして書き込まれ、また上述の追加プログラムの先頭アドレス(即ち、その追加プログラムの命令群をRAM 14に格納した際の第1番目の命令が格納されたアドレス)が該比較アドレスデータに対応したベクタ・アドレス・データとして所定のベクタデータ書込み領域(VAZ0、VAZ1、…VAZ6、VAZ7)に書き込まれ、更にそれら比較アドレスデータ及びベクタ・アドレス・データと対応関係にあるフラグデータ(FE0、FE1、…FE6、FE7)に“1”が与えられる。例えば、ROM 12のプログラムに追加プログラムを追加すべき箇所に対応したアドレスが比較アドレスデータCA1として比較アドレス書込み領域CAZ1に書き込まれると、その比較アドレス書込み領域CAZ1と対応関係にあるベクタアドレス書込み領域VAZ1には上述の追加プロ

グラムの先頭アドレスがベクタ・アドレス・データVA1として書き込まれ、その双方のアドレス書込み領域CAZ1及びVAZ1と対応関係にあるフラグデータFE1には“1”が与えられる。

【0042】

ROM12内のプログラムの第3のプログラム修正例として、ROM12のプログラムに含まれる一連の命令群（例えば、サブルーチン）をそのプログラムの別の箇所で新たに実行させるようにプログラムを変更することが必要となった場合には、その変更を行わせるプログラムはROM12に含まれる該サブルーチンを呼び出す形でEEPROM18の修正／追加プログラム書込み領域に変更プログラムとして書き込まれ、このときROM12内のプログラムに含まれる一連の命令群（該サブルーチン）を新たに実行させるべき箇所に対応したアドレスがEEPROM18の比較アドレス書込み領域(CAZ0、CAZ1、…CAZ6、CAZ7)のいずれかに比較アドレスデータとして書き込まれ、また上述の変更プログラムの先頭アドレス（即ち、その変更プログラムの命令群をRAM14に格納した際の第1番目の命令が格納されたアドレス）が該比較アドレスデータに対応したベクタ・アドレス・データとして所定のベクタデータ書込み領域(VAZ0、VAZ1、…VAZ6、VAZ7)に書き込まれ、更にそれら比較アドレスデータ及びベクタ・アドレス・データと対応関係にあるフラグデータ(FE0、FE1、…FE6、FE7)に“1”が与えられる。例えば、ROM12のプログラムに含まれる一連の命令群を新たに実行すべき箇所に対応したアドレスが比較アドレスデータCA2として比較アドレス書込み領域CAZ2に書き込まれると、その比較アドレス書込み領域CAZ2と対応関係にあるベクタアドレス書込み領域VAZ2には上述の変更プログラムの先頭アドレスがベクタ・アドレス・データVA2として書き込まれ、その双方のアドレス書込み領域CAZ2及びVAZ2と対応関係にあるフラグデータFE2には“1”が与えられる。

【0043】

なお、各フラグデータ(FE0、FE1、…FE6、FE7)はそれと対応関係にある比較アドレス書込み領域(CAZ0、CAZ1、…CAZ6、CAZ7)に書き込まれた比較アドレスデータが実際のプログラム修正に関与しているか否かを判別するためのフラグデータとして機能するものである。従って、上述の3つの修正例のように、比較ア

ドレスデータCA0、CA1及びCA2が実際のプログラム修正に関与すると、それらアドレスデータのそれぞれと対応関係にある3ビット分のフラグデータFE0、FE1及びFE2だけに“1”が与えられ、その他のフラグデータFE4ないしFE7については“0”の儘とされる。

【0044】

以上述べたような方法でROM12のプログラムに対して種々の修正が行われ、その修正情報データはすべてEEPROM18に保持されている。CPU10の立上げ時、16ビット構成の比較アドレスデータCA0、CA1、…CA6及びCA7はEEPROM18から読み出されて8つの回路素子 48_n のそれぞれの比較アドレスレジスタ50に書き込まれ、また16ビット構成のベクタ・アドレス・データVA0、VA1、…VA6及びVA7もEEPROM18から読み出されて8つの回路素子 48_n のそれぞれのベクタ・アドレス・レジスタ70に書き込まれ、更にはフラグデータFE0、FE1、…FE6及びFE7もEEPROM18から読み出されて8つの回路素子 48_n のそれぞれのアドレス一致割込み許可レジスタ68に書き込まれる。

【0045】

図4ないし図11を参照すると、各回路素子 48_n の比較アドレスレジスタ50、アドレス一致割込み許可レジスタ68及びベクタ・アドレス・レジスタ70のそれぞれに書き込まれた比較アドレスデータCA0、CA1、…CA6及びCA7、フラグデータFE0、FE1、…FE6及びFE7並びにベクタ・アドレス・データVA0、VA1、…VA6及びVA7が概念図として模式的に図示されている。上述の3つの修正例にあっては、図4ないし図6に示した回路素子 48_0 、 48_1 及び 48_2 の比較アドレスレジスタ50及びベクタ・アドレス・レジスタ70のそれぞれには実際のプログラム修正に関与した比較アドレスデータ(CA0、CA1、CA2)及びベクタ・アドレス・データ(VA0、VA1、VA2)が格納されている。一方、CPU10の立上げ時、図7ないし図11に示したその他の回路素子 48_3 ないし 48_7 の比較アドレスレジスタ50及びベクタ・アドレス・レジスタ70のそれぞれにもEEPROM18から比較アドレスデータ(CA3、CA4、…CA6、CA7)及びベクタ・アドレス・データ(VA3、VA4、…VA6、VA7)が読み出されて格納されることにな

るが、しかしそれらアドレスデータは実際のプログラム修正には関与するものでなく、このとき各アドレスデータには適当な値の16ビットデータ（通常は、すべてが“0”）が与えられることになる。しかしながら、各回路素子48_nのアドレス一致割込み許可レジスタ68に書き込まれたフラグデータ（FE0、FE1、…FE6、FE7）の値により、比較アドレスレジスタ50及びベクタ・アドレス・レジスタ70のそれぞれアドレスデータが実際のプログラム修正に関与したものであるかが確認され得る。

【0046】

一方、CPU10の立上げ時には、EEPROM18の修正／追加プログラム書込み領域から上述したような修正プログラム、追加プログラム及び変更プログラムが読み出されてRAM14に書き込まれ、このとき図1のマイクロコンピュータを搭載した電子機器に係わる種々のデータ等（補正データ、調整データ等）もEEPROM18のデータ書込み領域から読み出されてRAM14の所定領域に書き込まれる。

【0047】

図12を参照すると、RAM14のメモリ領域が概念図として模式的に図示され、同図に示すように、RAM14のメモリ領域にはプログラム実行作業領域、データ格納領域、修正／追加プログラム格納領域及びスタックメモリ領域が用意される。RAM14のプログラム実行作業領域及びスタックメモリ領域はCPU10でROM12のプログラムを実行する際に一時的なデータを保持したり処理待ちのデータ保持したりする作業領域として利用されるものであり、データ格納領域はEEPROM18のデータ書込み領域から読み出された種々のデータ等（補正データ、調整データ等）の格納領域として使用されるメモリ領域であり、修正／追加プログラム格納領域はEEPROM18の修正／追加プログラム書込み領域から読み出された修正プログラム、追加プログラム及び変更プログラムを格納するために使用されるメモリ領域である。

【0048】

図13ないし図20を参照すると、回路素子48₀ないし48₇のアドレス一致割込み一時禁止レジスタ58のそれぞれには1ビットデータとしてフラグデー

タFD0、FD1、…FD6及びFD7が定義され、これらフラグデータFD0、FD1、…FD6及びFD7の各々はCPU10の立上げ時にリセット信号により初期化されて“0”とされるが、各フラグデータ(FD0、FD1、…FD6、FD7)は後で詳述するような態様でCPU10の動作状況に応じて適宜書き換えられる。また、回路素子48₀ないし48₇のアドレス一致割込み要求クリアレジスタ74のそれぞれにも1ビットデータとしてフラグデータFC0、FC1、…FC6及びFC7が定義され、これらフラグデータFC0、FC1、…FC6及びFC7の各々もCPU10の立上げ時にリセット信号により初期化されて“0”とされるが、各フラグデータ(FC0、FC1、…FC6、FC7)は後で詳述するような態様でCPU10の動作状況に応じて適宜書き換えられる。更に、回路素子48₀ないし84₇のアドレス一致割込み監視レジスタ64のそれぞれにも1ビットデータとしてフラグデータFM0、FM1、…FM6及びFM7が定義され、これらフラグデータFM0、FM1、…FM6及びFM7の各々もCPU10の立上げ時にリセット信号により初期化されて“0”とされるが、各フラグデータ(FC0、FC1、…FC6、FC7)もCPU10の動作状況に応じて適宜書き換えられる。なお、回路素子48₀ないし48₇のアドレス一致割込み監視レジスタ64自体は本発明によるプログラム修正とは直接的には関係しないが、しかしその利用価値については後述する。

【0049】

CPU10の動作時、回路素子48₀ないし48₇のそれぞれのアドレス比較器52では、プログラムカウンタ26で逐次更新されるアドレスデータが順次取り込まれ、このアドレスデータはその該当比較アドレスレジスタ50に格納された比較アドレスデータ(CA0、CA1、…CA6、CA7)と比較される。プログラムカウンタ26からのアドレスデータが8つの比較アドレスレジスタ50内のいずれかのアドレスデータと一致したとき、例えば回路素子48₀の比較アドレスレジスタ50内の比較アドレスデータCA0がプログラムカウンタ26からのアドレスデータと一致すると、回路素子48₀のアドレス比較器52からはアドレス一致信号がハイレベル信号としてANDゲート54の一方の入力端子に対して出力される。一方、アドレス一致割込み一時禁止レジスタ58のフラグデータFD0には初期化により“0”が与えられ、このためアドレス一致割込み一時禁止レジスタ5

8からはローレベル信号がインバータ56に対して出力されているので、インバータ56の出力端子からはハイレベル信号がANDゲート54の他方の入力端子に対して出力される。

【0050】

従って、アドレス比較器52からアドレス一致割込み信号（ハイレベル）が出力されたときには、ANDゲート54の出力端子からはハイレベル信号がフリップフロップ62の入力端子Cに対して出力される。フリップフロップ62の入力端子には常にハイレベル信号が入力され、このためフリップフロップ62の出力端子Qからはアドレス一致割込み要求信号がハイレベル信号としてANDゲート66の一方の入力端子に対して出力される。

【0051】

ところで、上述した3つのプログラム修正例では、回路素子 48_0 、 48_1 及び 48_2 の比較アドレスレジスタ50に格納された比較アドレスデータCA0、CA1及びCA2は実際のプログラム修正に関与したものであるので、各比較アドレスデータ(CA0、CA1、CA2)と対応関係にあるアドレス一致割込み許可レジスタ68内のフラグデータ(FE0、FE1、FE2)には“1”が与えられており、このため上述の3つのプログラム修正例では回路素子 48_0 、 48_1 及び 48_2 のアドレス一致割込み許可レジスタ68だけからアドレス一致割込み許可信号がハイレベル信号としてANDゲート66の他方の入力端子に対して出力される。要するに、プログラムカウンタ26からのアドレスデータが回路素子 48_0 ないし 48_7 のいずれかの比較アドレスレジスタ50に格納された比較アドレスデータ(CA0、CA1、…CA6、CA7)と一致し、しかもその比較アドレスデータと対応関係にあるフラグデータ(FE0、FE1、…FE6、FE7)が“1”であるときだけ、ANDゲート66からの出力信号はローレベルからハイレベルに変化して、そのハイレベル信号（厳密にはその立上がり）がアドレス一致割込み信号として割込み制御部40とベクタ・アドレス・レジスタ70とに対して出力される。

【0052】

一方、CPU10の動作時にプログラムカウンタ26で逐次更新されるアドレスデータが実際のプログラム修正に関与していない比較アドレスデータと偶然一

致する場合があります。即ち、上述の3つのプログラム修正例では、回路素子48₃ ないし48₇ の比較アドレスレジスタ50に格納された比較アドレスデータCA3、CA4、…CA6 及びCA7 は実際のプログラム修正に関与していないが、しかしCPU10の立上げ時に回路素子48₃ ないし48₇ の比較アドレスレジスタ50のそれぞれには比較アドレスデータCA3、CA4、…CA6 及びCA7 として適当な16ビットデータが書き込まれるので、その16ビットデータがプログラムカウンタ26で逐次更新されるアドレスデータに偶然一致することがあり得る。勿論、そのような場合でも、アドレス比較器52からアドレス一致信号が出力され、フリップフロップ62の出力端子Qからアドレス一致割込み要求信号が出力されるが、しかしANDゲート66の出力端子からアドレス一致割込み信号が出力されることはない。というのは、実際のプログラム修正に関与していない比較アドレスデータ(CA3、CA4、…CA6、CA7)に対応関係にあるフラグデータ(FE3、FE4、…FE6、FE7)には“0”が与えられ、このためアドレス一致割込み許可レジスタ68からの出力信号はローレベルに維持されているからである。

【0053】

ANDゲート66からのアドレス一致割込み信号が割込み制御部40に入力されると、CPU10の制御ユニット20では、割込み移行処理が図21に示すような割込み移行シーケンスに従って実行され、この割込み移行処理の実行はROM12のプログラムとは無関係に割込み制御部40へのアドレス一致割込み信号の入力によって自動的に成される。なお、以下の説明では、上述の3つのプログラム修正例に倣って、回路素子48₀、48₁ 及び48₂ のいずれかのANDゲート66からアドレス一致割込み信号が割込み制御部40に対して出力された場合を想定している。

【0054】

先ず、ステップ2101では、プログラムカウンタ26に保持されているアドレス一致割込み終了後のリターンアドレス、即ちアドレス一致割込みが発生したアドレスがRAM14のスタックメモリ領域に一時的に退避される。次いで、ステップ2102では、状態レジスタ38のステータス情報(種々のフラグデータ)がRAM14のスタックメモリ領域に一時的に退避させられる。続いて、ステ

ップ 2103 では、状態レジスタ 38 に割り当てられた割込み許可フラグが“0”に書き換えられ、これにより緊急時の割込み処理 (NMI) 及びソフトウェア割込み処理以外の全ての割込み処理が禁止される。なお、以上で述べた処理は従来のマイクロコンピュータで実施される通常の割込み処理に含まれるものである。

【0055】

ステップ 2104 では、ベクタ・アドレス・レジスタ 70 からベクタ・アドレス・データ (VA0、VA1、VA2) がプログラムカウンタ 26 に設定される。なお、プログラムカウンタ 26 のアドレスデータを 8 つの回路素子 48_n のベクタ・アドレス・レジスタ 50 のうちのどのベクタ・アドレス・データ (VA0、VA1、…VA6、VA7) で更新設定するかについては、8 つの回路素子 48_n のうちのどのベクタ・アドレス・レジスタ 50 にアドレス一致割込み信号が入力されているかで決定され、上述の 3 つのプログラム修正では、回路素子 48_0 、 48_1 及び 48_2 のうちのいずれかのベクタ・アドレス・レジスタ 50 にアドレス一致割込み信号が入力されることになるので、その該当ベクタ・アドレス・データ (VA0、VA1、VA2) によってプログラムカウンタ 26 のアドレスデータが更新される。

【0056】

プログラムカウンタ 26 でアドレス一致割込みが発生したアドレスからベクタアドレス (VA0、VA1、VA2) への更新が行われると、修正プログラム、追加プログラム及び変更プログラムのいずれかの先頭アドレス (RAM 14 の領域) へ分岐し、かくして上述した修正プログラム、追加プログラム及び変更プログラムのいずれかが割込み処理として実行される (ステップ 2105)。

【0057】

上述した割込み処理では、まず、アドレス一致割込み要求クリアレジスタ 74 のフラグデータ (FC0、FC1、FC2) が“0”から“1”に書き換えられ、これによりアドレス一致割込み要求クリアレジスタ 74 からクリア信号 CL (ハイレベル) がリセット回路 72 に対して出力される。クリア信号 CL がリセット回路 72 に入力されると、リセット回路 72 では先ずリセット信号 RE2 (ハイレベル) が生成されてフリップフロップ 62 のリセット端子 R に対して出力され、これによりフリップフロップ 62 のラッチが解除されて、その出力端子 Q からのアドレ

ス一致割込み要求信号の出力が停止される。即ち、リセット端子 R へのリセット信号 RE 2 の出力により、フリップフロップ 6 2 の出力端子 Q からの出力信号がハイレベルからローレベルに戻される。かくして、割込み制御部 4 0 への AND ゲート 6 6 からのアドレス一致割込み信号の出力も停止される。続いて、リセット回路 7 2 からはリセット信号 RE 3 がアドレス一致割込み要求クリアレジスタ 7 4 に対して出力され、これによりアドレス一致割込み要求クリアレジスタ 7 4 のフラグデータ (FC0、FC1、FC2) が初期化されて、その値が “1” から “0” に戻される。

【0058】

図 2 2 を参照すると、リセット回路 7 2 の詳細回路図が示され、同図に示すように、リセット回路 7 2 には第 1 のフリップフロップ 7 2 A、第 2 のフリップフロップ 7 2 B、AND ゲート 7 2 C 及び OR ゲート 7 2 D が設けられる。アドレス一致割込み要求クリアレジスタ 7 4 から出力端子は第 1 のフリップフロップ 7 2 A の入力端子 D と AND ゲート 7 2 C の一方の入力端子とに接続され、AND ゲート 7 2 C の他方の入力端子は第 2 フリップフロップ 7 2 B の反転出力端子 Q (バー付) に接続される。第 1 のフリップフロップ 7 2 A の出力端子 Q は第 2 のフリップフロップ 7 2 B の入力端子 D に接続され、第 2 のフリップフロップ 7 2 B の出力端子 Q は OR ゲート 7 2 D の一方の入力端子に接続される。OR ゲート 7 2 D の他方の入力端子には CPU 1 0 の立上げ時にリセット信号 (ハイレベル) が入力されるだけで、その後は OR ゲート 7 2 D の他方の入力端子の入力レベルはローレベルに維持された儘とされる。また、CPU 1 0 の立上げ後、第 1 及び第 2 のフリップフロップ 7 2 A 及び 7 2 B のそれぞれの入力端子 C には所定周波数のシステムクロックパルスが常時入力される。なお、CPU 1 0 の立上げ時には、第 1 及び第 2 のフリップフロップ 7 2 A 及び 7 2 B のそれぞれのリセット端子 R にもリセット信号入力され、これにより第 1 及び第 2 のフリップフロップ 7 2 A 及び 7 2 B の初期化が行われる。

【0059】

図 2 3 には図 2 2 のリセット回路 7 2 の動作タイミングチャートが示され、この動作タイミングチャートを参照して、リセット回路 7 2 の動作を以下に説明す

る。

【0060】

アドレス一致割込み要求クリアレジスタ74からクリア信号（ハイレベル）がリセット回路72に対して出力されるまでは、第1のフリップフロップ72Aの入力端子Dの入力レベルはローレベルとなっているので、第1のフリップフロップ72Aの出力端子Qからの出力信号Q1はローレベルに維持され、このため第2のフリップフロップ72Bの出力端子Qからの出力信号Q2はローレベルに維持され、またその反転出力端子Q（バー付）からの反転出力信号Q2（バー付）はハイレベルに維持される。要するに、アドレス一致割込み要求クリアレジスタ74からクリア信号（ハイレベル）がリセット回路72に入力されるまでは、ANDゲート72C及びORゲート72Dのそれぞれの出力端子の出力レベルはローレベルに維持されている。

【0061】

アドレス一致割込み要求クリアレジスタ74からクリア信号（ハイレベル）がリセット回路72に対して出力されると、そのクリア信号はANDゲート72Cの一方の入力端子に入力され、このときANDゲート72Cの他方の入力端子には第2のフリップフロップ72Bの反転出力端子Q（バー付）からハイレベルの反転出力信号Q2（バー付）が入力されているので、ANDゲート72Cの出力端子の出力レベルはローレベルからハイレベルに変化して、そのハイレベル信号がリセット信号RE2としてフリップフロップ62のリセット端子Rに対して出力され、これによりフリップフロップ62の出力端子Qからのアドレス一致割込み要求信号の出力が停止される。

【0062】

一方、クリア信号が第1のフリップフロップ72Aの入力端子Dに入力された直後、システムクロックパルスの立上がり同期して、第1のフリップフロップ72Aの出力端子Qからの出力信号Q1はローレベルからハイレベルに変化し、このハイレベルの出力信号Q1が第2のフリップフロップ72Bの入力端子Dに入力される。第2のフリップフロップ72Bの入力端子Dにハイレベルの出力信号Q1が入力された直後、システムクロックパルスの立上がり同期して、第2

のフリップフロップ 7 2 B の出力端子 Q からの出力信号 Q 2 はローレベルからハイレベルに変化すると共に第 2 のフリップフロップ 7 2 B の反転出力端子 Q (バー付) からの出力信号 Q 2 (バー付) はハイレベルからローレベルに変化し、このとき AND ゲート 7 2 C の出力端子の出力レベルがハイレベルからローレベルに変化してリセット信号 R E 2 の出力が停止されると共に OR ゲート 7 2 D の出力端子からの出力信号がローレベルからハイレベルに変化し、このハイレベル信号がリセット信号 R E 3 としてアドレス一致割込み要求クリアレジスタ 7 4 に対して出力される。

【 0 0 6 3 】

リセット信号 R E 3 がアドレス一致割込み要求クリアレジスタ 7 4 に入力されると、そのフラグデータ (FC0、FC1、FC2) が “1” から “0” に戻され、このためアドレス一致割込み要求クリアレジスタ 7 4 からリセット回路 7 2 へのクリア信号 C L の出力がハイレベルからローレベルに変化する。クリア信号 C L がローレベルに変化した後、システムクロックパルスの立上がり同期して、第 1 のフリップフロップ 7 2 A のラッチが解除されて、その出力端子 Q からの出力信号 Q 1 はハイレベルからローレベルに戻る。また、第 1 のフリップフロップ 7 2 A の出力端子 Q からの出力信号 Q 1 がローレベルに戻った直後、システムクロックパルスの立上がり同期して、第 2 のフリップフロップ 7 2 B のラッチが解除されて、その出力端子 Q からの出力信号 Q 2 はハイレベルからローレベルに戻ると共にその反転出力端子 Q (バー付) からの反転出力信号 Q 2 (バー付) もローレベルからハイレベルに戻る。

【 0 0 6 4 】

要するに、アドレス一致割込み信号が AND ゲート 6 6 から割込み制御部 4 0 に対して出力されて、所定の割込み処理が行われるや否や、アドレス一致割込み要求クリアレジスタ 7 4 のフラグデータ (FC0、FC1、FC2) が “0” から “1” に書き換えられ、これにより AND ゲート 6 6 から割込み制御部 4 0 に対するアドレス一致割込み信号の出力が停止させられ、次いでアドレス一致割込み要求クリアレジスタ 7 4 のフラグデータ (FC0、FC1、FC2) が初期化されて “1” から “0” に戻される。

【0065】

以上で述べたようなプログラム修正例による割込み処理（上述の第1のプログラム修正例は除く）が完了した後に割込み時のアドレスに復帰する場合には、その割込み処理の完了前にアドレス一致割込み一時禁止レジスタ58のフラグデータ(FD1、FD2)が“0”から“1”に書き換えられ、このときアドレス一致割込み一時禁止レジスタ58の出力信号がローレベルからハイレベルに変化させられて、インバータ56からの出力信号がローレベルとされる。次いで、かかる割込み処理が完了すると、修正・追加プログラム最後の割込み復帰命令に基づいて、CPU10の制御ユニット20では、割込み復帰処理が図24に示すような割込み復帰シーケンスに従って実行される。

【0066】

ステップ2401では、RAM14のスタックメモリ領域に一時的に退避されていたステータス情報（種々のフラグデータ）が状態レジスタ38に戻され、このとき割込み許可フラグも割込み発生前の状態に戻される。次いで、ステップ2402ではRAM14のスタックメモリ領域に退避されていたリターンアドレスデータがプログラムカウンタ26に戻されて設定される。即ち、プログラムカウンタ26のアドレスが割込み発生時のアドレスによって更新される。続いて、ステップ2403でプログラムカウンタ26のアドレス（割込み発生時のアドレス）に復帰する。かくして、修正プログラムが終了し、アドレス一致割込みの発生で中断していた処理がROM12のプログラムに従って続行される。

【0067】

ところで、上述の割込み復帰シーケンスでプログラムカウンタ26のアドレスが割込み発生時のリターンアドレスデータによって更新されたとき（ステップ2402）、その更新アドレスデータは比較アドレスレジスタ50の比較アドレスデータ(CA1、CA2)と一致し、このためアドレス比較器52から再度アドレス一致信号が出力されることになる。しかしながら、そのときには、上述したように割込み処理の完了前にアドレス一致割込み一時禁止レジスタ58のフラグデータ(FD1、FD2)が“1”とされ、このためインバータ56の出力信号はローレベルとされている。従って、ANDゲート54の出力信号レベルはローレベルされるので

、フリップフロップ62の出力端子Qからアドレス一致割込み要求信号が出力されることなく、このためCPU10の制御ユニット20ではプログラムカウンタ26で更新されたリターンアドレスにアクセスしてROM12のプログラムに従う所定の処理が続行される。

【0068】

一方、上述の割込み復帰シーケンスでプログラムカウンタ26のアドレスが割込み時のリターンアドレスデータによって更新された際にアドレス比較器52からハイレベルのアドレス一致信号が再度出力される。ROM12のプログラムが実行され、次のステップに移り、プログラムカウンタ26が更新され、アドレス一致信号がハイレベルからローレベルに変化すると、リセット回路60からリセット信号RE1がアドレス一致割込み一時禁止レジスタ58に対して出力され、これによりアドレス一致割込み一時禁止レジスタ58は初期化されて“1”から“0”に戻される。

【0069】

図25を参照すると、リセット回路60の詳細回路図が示され、同図に示すように、リセット回路60には第1のフリップフロップ60A、第2のフリップフロップ60B、インバータ60C、ANDゲート60D及びORゲート60Eが設けられる。アドレス比較器52の出力端子は第1のフリップフロップ60Aの入力端子Dとインバータ60Cの入力端子とに接続され、第1のフリップフロップ60Aの出力端子Qは第2のフリップフロップ60Bの入力端子Dに接続され、インバータ60Cの出力端子はANDゲート60Dの一方の入力端子に接続され、ANDゲート60Dの他方の入力端子は第2のフリップフロップ60Bの出力端子Qに接続される。ANDゲート60Dの出力端子はORゲート60Eの一方の入力端子に接続され、ORゲート60Eの他方の入力端子にはCPU10の立上げ時にリセット信号（ハイレベル）が入力されるだけで、その後はORゲート60Eの他方の入力端子の入力レベルはローレベルに維持された儘とされる。また、CPU10の立上げ後、第1及び第2のフリップフロップ60A及び60Bのそれぞれの入力端子Cには所定周波数のシステムクロックパルスが常時入力される。なお、CPU10の立上げ時には、第1及び第2のフリップフロップ

60A及び60Bのそれぞれのリセット端子Rにもリセット信号が入力され、これにより第1及び第2のフリップフロップ60A及び60Bの初期化が行われる。

【0070】

図26には図25のリセット回路60の動作タイミングチャートが示され、この動作タイミングチャートを参照して、リセット回路60の動作について以下に説明する。

【0071】

図26の動作タイミングチャートから明らかなように、CPU10の立上げ時にリセット回路60は初期化され、このときリセット回路60の出力端子の出力レベル(RE1)はローレベルに維持されている。上述の割込み復帰シーケンスでプログラムカウンタ28のアドレスが割込み時のリターンアドレスデータに更新されてアドレス比較器52からアドレス一致信号(ハイレベル)がリセット回路60に入力されたとき、先ずインバータ60Cの出力端子の出力信号INVがハイレベルからローレベルに変化し、そのローレベル信号はANDゲート60Dの一方の入力端子に入力される。

【0072】

一方、アドレス一致信号(ハイレベル)が第1のフリップフロップ60Aの入力端子Dに入力された直後、所定周波数のシステムクロックパルスの立上がり同期して、第1のフリップフロップ60Aの出力端子Qからの出力信号Q1はローレベルからハイレベルに立ち上がり、このハイレベルの出力信号Q1が第2のフリップフロップ60Bの入力端子Dに入力された直後、所定周波数のシステムクロックパルスの立上がり同期して、第2のフリップフロップ72Bの出力端子Qからの出力信号Q2はローレベルからハイレベルに立ち上がり、このハイレベル信号はANDゲート60Dの他方の入力端子に入力される。しかしながら、ANDゲート60Dの一方の入力端子に入力される入力信号(インバータ60Cからの)はローレベルとなっているために、ANDゲート60Dからの出力信号ANDはローレベルとなっている。

【0073】

ROM 12 のプログラムが実行されて、プログラムカウンタ 28 のアドレスデータが次のアドレスデータに更新されると、その更新アドレスデータは比較アドレスレジスタ 50 の比較アドレスデータ (CA1、CA2) とは一致しなくなるので、アドレス比較器 52 からのアドレス一致信号の出力は停止される (即ち、アドレス比較器 52 からの出力信号はハイレベルからローレベルに変化する)。アドレス一致信号の出力が停止されると、インバータ 60C の出力信号はローレベルからハイレベルに変化し、このため AND ゲート 60D の出力信号 AND はローレベルからハイレベルとなって、OR ゲート 60E からはリセット信号 RE1 がハイレベル信号としてアドレス一致割込み一時禁止レジスタ 58 に対して出力されるので、アドレス一致割込み一時禁止レジスタ 58 がリセットされて、そのフラグデータ (FD1、FD2) が “1” から “0” に戻される。

【0074】

一方、アドレス一致信号の出力停止の直後、システムクロックパルスの立上がりに同期して、第 1 のフリップフロップ 60A のラッチが解除され、これにより第 1 のフリップフロップ 60A の出力端子 Q からの出力信号 Q1 はハイレベルからローレベルに変化し、そのローレベルの出力信号 Q1 は第 2 のフリップフロップ 60B の入力端子 D に入力される。ローレベルの出力信号 Q1 が第 2 のフリップフロップ 60B の入力端子 D に入力された直後、システムクロックパルスの立上がりに同期して、第 2 のフリップフロップ 60B のラッチが解除され、これにより第 2 のフリップフロップ 60B の出力端子 Q からの出力信号 Q2 はハイレベルからローレベルに変化し、これにより AND ゲート 60D の出力信号はハイレベルからローレベルに変化すると共に OR ゲート 60E からのリセット信号 RE1 (ハイレベル) の出力が停止される。

【0075】

要するに、リセット信号 RE1 の出力によりアドレス一致割込み一時禁止レジスタ 58 のフラグデータ (FD1、FD2) が “1” から “0” に戻されるので、その後の ROM 12 のプログラムに従う処理の実行中、比較アドレスレジスタ 50 の比較アドレスデータ (CA1、CA2) が再びプログラムカウンタ 26 のアドレスデータと一致してアドレス比較器 52 からアドレス一致信号が出力されたときには、AN

Dゲート54からハイレベルの信号が出力され、これによりフリップフロップ62の出力端子Qからはアドレス一致要求信号が出力されるので、上述した割込み処理（第2及び第3のプログラム修正例）が再度実行される。

【0076】

上述した第1のプログラム修正による割込み処理の場合には、アドレス一致割込み一時レジスタ58のフラグデータFD0の“0”から“1”への書換えは行われず、また図24の割込み復帰シーケンスも実行されない。というのは、もし第1のプログラム修正による割込み処理の完了後に割込み時のアドレスに復帰すると、所謂バグを含む一連の命令群（ROM12内）が実行されてしまうことになるからである。このような場合には、かかる割込み処理の完了時に別のアドレスに分岐させられる。即ち、第1のプログラム修正による割込み処理の完了時には、プログラムカウンタ26にはその分岐先アドレスが設定され、CPU10の制御ユニット20では該分岐先アドレスに従って処理が続行される。勿論、そのような場合には、割込み移行シーケンスでRAM14のスタックメモリ領域に退避されていた状態レジスタステータス情報は修正プログラムで適宜復帰されるか或いは廃棄され、またそのリターンアドレスは廃棄される。なお、上述した第2及び第3のプログラム修正例による割込み処理の完了後に必ずしも割込み時のアドレスに復帰する必要はなく、必要に応じてその他のアドレスに分岐してよく、その場合には第1のプログラム修正例の場合と同様な処理が行われる。

【0077】

図27を参照すると、本発明によるマイクロコンピュータをプリンタ、ビデオカメラ、カメラや時計等の電子機器に搭載した際にCPU10で実行されるシステム作動ルーチンのフローチャートの一例が示されている。なお、このシステム作動ルーチンは上述したような電子機器の電源スイッチのオンによって実行される。

【0078】

先ず、ステップ2701では、CPU10の初期化が行われ、次いでステップ2702で上述した電子機器内の周辺回路（即ち、本発明によるマイクロコンピュータを搭載した電子機器内の回路システム等）の初期化が行われる。

【0079】

ステップ2703では、EEPROM18に対するプログラム及びデータ等の書換え要求があるか否かが判断される。上述したように、EEPROM18がマイクロコンピュータの基板に恒久的に固着された状態で搭載されているとき、EEPROM18へのプログラム及びデータ等の書込みは図1のマイクロコンピュータをEEPROM書込み用コンピュータに接続させることにより行われ、この場合には上述の電子機器の電源オン（即ち、CPU10の立上げ）前にマイクロコンピュータはEEPROM書込み用コンピュータに接続され、該EEPROM書込み用コンピュータからCPU10に対して書換え要求信号が出力され、これによりCPU10ではその立上げ後にEEPROM書込み用コンピュータからの書換え要求信号の出力の有無が判断される。

【0080】

ステップ2703でEEPROM書込み用コンピュータからの書換え要求信号が出力されていることが確認されると、ステップ2704に進み、そこでROM12に用意されていたEEPROM書込みプログラムに従ってEEPROM書換えルーチンが実行され、これによりEEPROM書込み用コンピュータの制御下でEEPROM18に修正プログラムや追加プログラム及びアドレス一致割込み関係のデータ（即ち、比較アドレスデータ(CA0、CA1、…CA6、CA7)、ベクタ・アドレス・データ(VA0、VA1、…VA6、VA7)、フラグデータ(FE0、FE1、…FE6、FE7))が書き込まれる。

【0081】

勿論、EEPROM18への修正プログラムや追加プログラム及び所定のフラグデータの書込みが一旦完了している場合、或いは更なるプログラムの修正が必要とされない場合やプログラム修正とは無関係なデータ等の書込みが必要とされない場合、CPU10がEEPROM書込み用コンピュータに接続されることはなく、その場合にはステップ2703からステップ2704を迂回してステップ2705に直ちに進む。何れにしても、ステップ2705では、プログラム修正とは無関係なデータ、例えば上記電子機器に関する補正／調整データ等がEEPROM18から読み出されてRAM14NIに取り込まれる。なお、EEPROM

1 8 が取外し自在となっているときには、その E E P R O M 1 8 への書込みは E E P R O M 書込み用コンピュータによって直接行うことができるので、ステップ 2 7 0 3 及び 2 7 0 4 は必要とされない。

【 0 0 8 2 】

ステップ 2 7 0 6 では、E E P R O M 1 8 内の修正／追加プログラムの読出しバイト数が零であるか否かが判断される。もし E E P R O M 1 8 内に修正／追加プログラムが格納されている場合には、E E P R O M 1 8 ではその修正／追加プログラムに応じたバイト数が使用されているが、しかしもしその使用バイト数が零である場合には、R O M 1 2 の格納プログラムに対する修正はなく、E E P R O M 1 8 には修正／追加プログラムが格納されていないということになる。

【 0 0 8 3 】

E E P R O M 1 8 内に修正／追加プログラムが格納されているとき、即ち E E P R O M 1 8 内の修正／追加プログラムの読出しバイト数が零でないとき、ステップ 2 7 0 7 に進み、そこで E E P R O M 1 8 から修正／追加プログラムが所定バイト数だけ読み出されて R A M 1 4 に取り込まれる。

【 0 0 8 4 】

次いで、ステップ 2 7 0 8 では、E E P R O M 1 8 から比較アドレスデータ C A 0 、 C A 1 、 … C A 6 及び C A 7 が読み出されて回路素子 4 8 0 ないし 4 8 7 のそれぞれの比較アドレスレジスタ 5 0 に設定され、また E E P R O M 1 8 からベクタ・アドレス・データ V A 0 、 V A 1 、 … V A 6 及び V A 7 が読み出されて回路素子 4 8 0 ないし 4 8 7 のそれぞれの B ベクタ・アドレス・レジスタ 7 0 に設定される。更に、E E P R O M 1 8 からフラグデータ F E 0 、 F E 1 、 … F E 6 及び F E 7 も読み出されて回路素子 4 8 0 ないし 4 8 7 のそれぞれのアドレス一致割込み許可レジスタ 6 8 に設定される。

【 0 0 8 5 】

その後、ステップ 2 7 0 9 では、C P U 1 0 が組み込まれた電子機器に応じた所定の処理ルーチンが R O M 1 2 内のプログラムに従って実行される。一方、E E P R O M 1 8 内に修正／追加プログラムが格納されていないとき、即ち E E P R O M 1 8 内の修正／追加プログラムの読出しバイト数が零であるとき、ステッ

ブ 2707 及び 2708 を迂回してステップ 2709 に直ちに進む。

【0086】

図 28 を参照すると、図 27 のステップ 2709 で実行されるべき処理ルーチンの一部が例示的に示される。同図から明らかなように、この例では、処理ルーチンはステップ 2801 ないし 2810 から成り、各ステップでは所定のアドレス (ROM 12 の領域) に格納された命令が実行される。換言すれば、図 28 の処理ルーチンはプログラム化されて ROM 12 の領域で所定アドレスに格納される。なお、図 1 に示すマイクロコンピュータでは、一命令は 4 バイト構成とされるので、各ステップで実行されるべき命令は 4 バイト分のアドレスに格納されたものとなっている。

【0087】

図 29 に示すように、ステップ 2801 では、サブルーチンとして構成された処理ルーチン A がコールされて実行されることになり、この命令は例えば ROM 12 の領域の 4 バイト分のアドレス 9800H ないし 9803H に格納されているものとされる。ステップ 2802 では、条件分岐 X を判別する命令が実行され、この命令は上述の 4 バイト分のアドレス 9800H ないし 9803H に続く次の 4 バイト分のアドレス 9804H ないし 9808H に格納される。なお、アドレス番号に付された H はそのアドレス番号が 16 進数であることを表している。

【0088】

ステップ 2802 での判別の結果として、ステップ 2802 からステップ 2803 に進むとすると、ステップ 2803 では、サブルーチンとして構成された処理ルーチン B がコールされて実行されることになり、この命令はアドレス 9808H ないし 980BH に格納される。次いで、ステップ 2804 では、サブルーチンとして構成された処理ルーチン C がコールされて実行されることになり、この命令はアドレス 980CH ないし 980FH に格納される。なお、ステップ 2804 で処理ルーチン C の実行が完了すると、図 28 の処理ルーチンを抜ける命令、即ち別の任意のステップ W にジャンプする命令が実行され、この命令はアドレス 9810H ないし 9813H に格納される。

【0089】

ステップ 2802 での判別の結果として、ステップ 2802 からステップ 2805 に進む場合には、アドレス 9814H にジャンプさせられる。ステップ 2805 では、条件分岐 Y を判別する命令が実行され、この命令はアドレス 9814H ないし 9817H に格納される。

【0090】

ステップ 2805 での判別の結果として、ステップ 2805 からステップ 2806 に進むとすると、ステップ 2806 では、サブルーチンとして構成された処理ルーチン D がコールされて実行されることになり、この命令はアドレス 9818H ないし 981BH に格納される。次いで、ステップ 2807 では、条件分岐 Z を判別する命令が実行され、この命令はアドレス 981CH ないし 981FH に格納される。

【0091】

ステップ 2807 での判別の結果として、ステップ 2807 からステップ 2808 に進むとすると、ステップ 2808 では、サブルーチンとして構成された処理ルーチン E がコールされて実行されることになり、この命令はアドレス 9820H ないし 9823H に格納される。なお、ステップ 2808 で処理ルーチン E の実行が完了すると、図 28 の処理ルーチンを抜ける命令、即ち別の任意のステップ W にジャンプする命令が実行され、この命令はアドレス 9824H ないし 9827H に格納される。

【0092】

ステップ 2807 での判別の結果として、ステップ 2807 からステップ 2809 に進む場合には、アドレス 9828H にジャンプさせられる。ステップ 2809 では、サブルーチンとして構成された処理ルーチン F がコールされて実行されることになり、この命令はアドレス 9828H ないし 982BH に格納される。なお、ステップ 2809 で処理ルーチン F の実行が完了すると、図 28 の処理ルーチンを抜ける命令、即ち別の任意のステップ W にジャンプする命令が実行され、この命令はアドレス 982CH ないし 982FH に格納される。

【0093】

ステップ 2805 での判別の結果として、ステップ 2805 からステップ 2810 に進む場合には、アドレス 9830H にジャンプさせられる。ステップ 2810

では、サブルーチンとして構成された処理ルーチンGがコールされて実行されることになり、この命令はアドレス9830H ないし9833H に格納される。なお、ステップ2810で処理ルーチンGの実行が完了すると、図28の処理ルーチンを抜ける命令、即ち別の任意のステップWにジャンプする命令が実行され、この命令はアドレス9834H ないし9838H に格納される。

【0094】

図29を参照すると、図28に示す処理ルーチンの各ステップでの命令とプログラムカウンタ26での更新アドレスとの関係が模式的に示されている。図28に示す処理ルーチンが実行されるとき、図1のマイクロコンピュータでは、制御ユニット20がROM12のプログラムに従って4バイト分のアドレスに書き込まれた命令を解読する間、プログラムカウンタ26では、1バイト分のアドレスが順次更新される。例えば、アドレス9800H ないし9803H が制御ユニット20により順次アクセスされて、処理ルーチンAのコール実行命令が解読される間、プログラムカウンタ26では、アドレス9800H から9803H までが順次更新される。要するに、図28に示す事例では、先ず、処理ルーチンAが実行され、その後の処理ルーチンB、C、D、F及びGの実行は条件分岐X、Y及びZに従って選択的に実行され間、プログラムカウンタ26では、図29に示すような態様でアドレスが1バイトずつ更新される。

【0095】

一方、図28のステップ2801、2803、2804、2806、2808、2809及び2810で実行される処理ルーチンA、B、C、D、E、F及びGは図30に示すようにROM12の領域にプログラム化されて所定アドレスに格納される。例えば、処理ルーチンAはその先頭アドレス1000H 以降のアドレスに一連の命令群から成るプログラムとしてROM12内に格納され、処理ルーチンBはその先頭アドレス2000H 以降のアドレスに一連の命令群から成るプログラムとしてROM12内に格納され、処理ルーチンCはその先頭アドレス3000H 以降のアドレスに一連の命令群から成るプログラムとしてROM12内に格納され、処理ルーチンDはその先頭アドレス4000H 以降のアドレスに一連の命令群から成るプログラムとしてROM12内に格納され、処理ルーチンEはその先頭アド

レス5000H 以降のアドレスに一連の命令群から成るプログラムとしてROM 1 2 内に格納され、処理ルーチンFはその先頭アドレス6000H 以降のアドレスに一連の命令群から成るプログラムとしてROM 1 2 内に格納され、処理ルーチンGはその先頭アドレス7000H 以降のアドレスに一連の命令群から成るプログラムとしてROM 1 2 内に格納される。

【0 0 9 6】

図 2 8 に示す事例において、もし処理ルーチンBに所謂バグが発見された場合にROM 1 2 のプログラムを修正する場合（即ち、上述の第 1 のプログラム修正例の場合）には、その処理ルーチンBのプログラムに代わる正常なプログラムが修正プログラムとして作成され、その後EEPROM 1 8 の修正／追加プログラム書込み領域に書き込まれる（ステップ 2 7 0 4）。このとき処理ルーチンBのコール命令が格納されたアドレス（4 バイト）の最初の一バイト分のアドレス98 08H が例えば比較アドレスデータCA0 [CA0₁₅, CA0₁₄, …CA0₀₁, CA0₀₀] としてEEPROM 1 8 のアドレスデータ書込み領域CAZ0に書き込まれ、またアドレスデータ書込み領域CAZ0と対応関係にあるアドレスデータ書込み領域VAZ0には修正プログラムの先頭アドレス（例えば、0600H）がベクタ・アドレス・データVA0 [VA0₁₅, VA0₁₄, …VA0₀₁, VA0₀₀] として書き込まれ、更にアドレスデータ書込み領域CAZ0と対応関係にあるフラグデータFE0 には“1”が与えられる（図 3 及び図 4）。

【0 0 9 7】

また、図 2 8 に示す事例において、もし処理ルーチンDの直前で新たな追加処理ルーチンの実行が必要となった場合（以下、第 2 のプログラム修正例）には、その追加処理ルーチンを実行するプログラムが追加プログラムとして作成され、その後EEPROM 1 8 の修正／追加プログラム書込み領域に書き込まれる（ステップ 2 7 0 4）。このとき処理ルーチンDのコール命令が格納されたアドレス（4 バイト）の最初の一バイト分のアドレス9818H が例えば比較アドレスデータCA1 [CA1₁₅, CA1₁₄, …CA1₀₁, CA1₀₀] としてEEPROM 1 8 のアドレスデータ書込み領域CAZ1に書き込まれ、またアドレスデータ書込み領域CAZ1と対応関係にあるアドレスデータ書込み領域VAZ1には追加プログラムの先頭アドレス（例えば

、0800H) がベクタ・アドレス・データVA1 [VA1₁₅, VA1₁₄, …VA1₀₁, VA1₀₀] として書き込まれ、更にアドレスデータ書込み領域CAZ1と対応関係にあるフラグデータFE1 には“1”が与えられる(図3及び図5)。

【0098】

更に、図28に示す事例において、もし処理ルーチンEの直前で処理ルーチンG (ROM12の領域) の実行が必要となった場合(以下、第3のプログラム修正例) には、その変更処理ルーチンを実行するプログラムが変更プログラムとして作成され、その後EEPROM18の修正/追加プログラムに書込み領域に書き込まれる(ステップ2704)。このとき処理ルーチンEのコール命令が格納されたアドレス(4バイト)の最初の一バイト分のアドレス9820H が例えば比較アドレスデータCA2 [CA2₁₅, CA2₁₄, …CA2₀₁, CA2₀₀] としてEEPROM18のアドレスデータ書込み領域CAZ2に書き込まれる。また、アドレスデータ書込み領域CAZ2と対応関係にあるアドレスデータ書込み領域VAZ2には変更プログラムの先頭アドレス(例えば、0A00H) がベクタ・アドレス・データVA2 [VA2₁₅, VA2₁₄, …VA2₀₁, VA2₀₀] として書き込まれ、更にアドレスデータ書込み領域CAZ1と対応関係にあるフラグデータFE2 には“1”が与えられる(図3及び図6)。

【0099】

既に述べたように、図27に示すシステム作動ルーチンの実行時、上述したような修正プログラム、追加プログラム及び変更プログラムはEEPROM18から読み出されてRAM14の修正/追加プログラム格納領域に取り込まれる(ステップ2707)。図31を参照すると、かかる修正プログラム、追加プログラム及び変更プログラムをRAM14に格納した状態が概念図として模式的に示され、同図から明らかなように、修正プログラムは割込み処理ルーチンJを実行するものとして先頭アドレス0600H 以降に格納され、また追加プログラムは割込み処理ルーチンKを実行するものとして先頭アドレス0800H 以降に格納され、更に変更プログラムは割込み処理ルーチンLを実行するものとして先頭アドレス0A00H 以降に格納される。なお、ここで言う先頭アドレスとは該当割込み処理ルーチンを実行する命令群が格納された先頭アドレスである。

【0100】

また、既に述べたように、図 27 に示すシステム作動ルーチンの実行時には、比較アドレスデータ CA0、CA1、…CA6 及び CA7、ベクタ・アドレス・データ VA0、VA1、…VA6 及び VA7、並びにフラグデータ FE0、FE1、…FE6 及び FE7 が回路素子 48₀ ないし 48₇ のそれぞれの比較アドレスレジスタ 50、ベクタ・アドレス・レジスタ 70 及びアドレス一致割込み許可レジスタ 68 に設定される（ステップ 2708）。なお、以上で説明した第 1 ないし第 3 のプログラム修正例では、実際のプログラム修正に関与しているアドレス一致割込み許可レジスタ 68 は回路素子 48₀、48₁ 及び 48₂ に含まれるものだけであるから、フラグデータ FE0、FE1 及び FE2 だけに“1”が与えられ、その他のフラグデータ FE3 ないし FE7 には“0”が与えられる。

【0101】

次に、上述した第 1 のプログラム修正例（修正プログラム）に従う CPU10 の動作について以下に説明する。

【0102】

まず、CPU10 の動作中、プログラムカウンタ 26 でアドレスが逐次更新され、その更新アドレスが処理ルーチン B のコール命令の格納されたアドレス（4 バイト）の最初の一バイト分のアドレス 9808H に到達すると、その更新アドレス 9808H は回路素子 48₀ の比較アドレスレジスタ 50 に格納された比較アドレスデータ [9808H] と一致することとなる。先に述べたように、双方のアドレスデータが互いに一致すると、回路素子 48₀ のアドレス比較器 52 からはアドレス一致信号（ハイレベル）が出力され、これによりフリップフロップ 62 の出力端子 Q からはアドレス一致割込み要求信号（ハイレベル）が出力され、このとき回路素子 48₀ のアドレス一致割込み許可レジスタのフラグデータ FE0 には“1”が与えられているので、回路素子 48₀ の AND ゲート 66 からはアドレス一致割込み信号が割込み制御部 40 に対して出力される。

【0103】

回路素子 48₀ の AND ゲート 66 から出力されたアドレス一致割込み信号が割込み制御部 40 に出力されると、制御ユニット 20 では、処理ルーチン B のコール命令が格納されたアドレス（9808H～980BH）にアクセスすることなく割込み移

行処理が上述したような割込み移行シーケンス（図 21）に従って実行される。即ち、プログラムカウンタ 26 に保持されているアドレスデータ 9808H が RAM 14 のスタックメモリ領域に一時的に退避されると共にその時点での状態レジスタ 38 のステータス情報（種々のフラグデータ）も RAM 14 のスタックメモリ領域に一時的に退避させられる。次いで、プログラムカウンタ 26 には回路素子 48₀ のベクタ・アドレス・レジスタ 70 内のベクタ・アドレス・データ 0600H が設定されて、RAM 14 の割込み処理ルーチン J の先頭アドレス 0600H に分岐し、割込み処理ルーチン J が実行される。なお、割込み移行処理時には、先に述べたように状態レジスタ 38 の割込み許可フラグが“0”とされ、これにより緊急時の割込み処理（NMI）及びプログラムによる割込み処理以外の全割込み処理が禁止される。

【0104】

割込み処理ルーチン J では、まず、ステップ J1 で回路素子 48₀ のアドレス一致割込み要求クリアレジスタ 74 のフラグデータ FC0 が“0”から“1”に書き換えられ、これによりクリア信号（ハイレベル）がアドレス一致割込み要求クリアレジスタ 74 からリセット回路 72 に対して出力される。このクリア信号がアドレス一致割込み要求クリアレジスタ 74 に対して出力されると、リセット回路 72 からはリセット信号 RE2（ハイレベル）が先に説明したような態様でフリップフロップ 62 のリセット端子 R に対して出力され（図 22 及び図 23）、これによりフリップフロップ 62 のラッチ状態が解除されて、その出力端子 Q からのアドレス一致割込み要求信号の出力が停止される。また、リセット回路 72 からはリセット信号 RE2 に続いてリセット信号 RE3 がアドレス一致割込み要求クリアレジスタ 74 に対して出力され、これによりアドレス一致割込み要求クリアレジスタのフラグデータ FC0 が“1”から“0”に書き戻される。

【0105】

ステップ J2 では、バグを含むプログラム（処理ルーチン B）に代わる修正プログラム（修正ルーチン）が実行され、修正ルーチンの実行が完了すると、ステップ J3 に進み、そこで RAM 14 のスタックメモリ領域に一時的に退避されていた状態レジスタ 38 のステータス情報等が復帰させられると共にリターンアド

レスが廃棄される。次いで、ステップ J 4 では、処理ルーチン C のコール命令が格納されているアドレス（4 バイト）の最初の一バイトのアドレス 980CH にジャンプさせられ、これにより制御ユニット 20 ではアドレス 980CH がアクセスされて、処理ルーチン C が実行される。

【0106】

以上の記載から明らかなように、本発明によれば、ROM 12 のプログラムを構成する処理ルーチン B に所謂バグが発見されたとしても、その処理ルーチン B に代わる割込み処理ルーチン J の実行により、ROM 12 のプログラム全体は正常に機能され得ることになる。

【0107】

次に、上述した第 2 のプログラム修正例（追加プログラム）に従う CPU 10 の動作について以下に説明する。

【0108】

上述の第 1 のプログラム修正例と同様に、プログラムカウンタ 26 の更新アドレスが処理ルーチン D のコール命令の格納されたアドレス（4 バイト）の最初の一バイト分のアドレス 9818H に到達すると、その更新アドレス 9818H は回路素子 48₁ の比較アドレスレジスタ 50 に格納された比較アドレスデータ [9818H] と一致することとなる。先に述べたように、双方のアドレスデータが互いに一致すると、回路素子 48₁ のアドレス比較器 52 からはアドレス一致信号（ハイレベル）が出力され、これによりフリップフロップ 62 の出力端子 Q からはアドレス一致割込み要求信号（ハイレベル）が出力され、このとき回路素子 48₁ のアドレス一致割込み許可レジスタのフラグデータ FE1 には“1”が与えられているので、回路素子 48₁ の AND ゲート 66 からはアドレス一致割込み信号が割込み制御部 40 に対して出力される。

【0109】

回路素子 48₁ の AND ゲート 66 から出力されたアドレス一致割込み信号が割込み制御部 40 に出力されると、制御ユニット 20 では、処理ルーチン D のコール命令が格納されたアドレス（9818H～981BH）にアクセスすることなく割込み移行処理が上述したような割込み移行シーケンス（図 21）に従って実行される。

即ち、プログラムカウンタ 26 に保持されているアドレスデータ 9818H が RAM 14 のスタックメモリ領域に一時的に退避されると共にその時点での状態レジスタ 38 のステータス情報（種々のフラグデータ）も RAM 14 のスタックメモリ領域に一時的に退避させられる。次いで、プログラムカウンタ 26 には回路素子 48_1 のベクタ・アドレス・レジスタ 70 内のベクタ・アドレス・データ 0800H が設定されて、RAM 14 の割込み処理ルーチン K の先頭アドレス 0800H に分岐し、割込み処理ルーチン K が実行される。勿論、上述の場合と同様に、割込み移行処理時には、状態レジスタ 38 の割込み許可フラグが“0”とされ、これにより緊急時の割込み処理（NMI）及びプログラムによる割込み処理以外の全割込み処理が禁止される。

【0110】

割込み処理ルーチン K では、まず、ステップ K 1 で回路素子 48_1 のアドレス一致割込み要求クリアレジスタ 74 のフラグデータ FC1 が“0”から“1”に書き換えられ、これによりクリア信号（ハイレベル）がアドレス一致割込み要求クリアレジスタ 74 からリセット回路 72 に対して出力される。上述の場合と同様に、このクリア信号の出力により、これにより回路素子 48_1 のフリップフロップ 62 のラッチ状態が解除されて、その出力端子 Q からのアドレス一致割込み要求信号の出力が停止され、またアドレス一致割込み要求クリアレジスタのフラグデータ FC1 が“1”から“0”に書き戻される。

【0111】

ステップ K 2 では、追加プログラム（追加ルーチン）が実行され、追加ルーチンの実行が完了すると、ステップ K 3 に進み、そこでアドレス一致割込み一時禁止レジスタ 58 のフラグデータ FD1 が“0”から“1”に書き換えられる。

【0112】

その後、追加プログラムでの最後の割込み処理からの復帰（リターン）命令により、割込み復帰処理が上述したような割込み復帰シーケンス（図 24）に従って実行される。即ち、RAM 14 のスタックメモリ領域に一時的に退避されていたステータス情報が状態レジスタ 38 に戻され、また RAM 14 のスタックメモリ領域に一時的に退避されていた割込み時のアドレスデータ 9818H がプログラム

カウンタ 2 6 に再び設定される。

【0 1 1 3】

ところで、割込み処理ルーチン K の復帰（リターン）時、プログラムカウンタ 2 6 には上述したように R A M 1 4 のスタックメモリ領域に一時的に退避されていた割込み発生時のアドレスデータ 9818H が設定されるために、そのアドレスデータ 9818H は回路素子 4 8₁ の比較アドレスレジスタ 5 0 内の比較アドレスデータと再び一致することとなり、このため回路素子 4 8₁ のアドレス比較器 5 2 からはアドレス一致信号（ハイレベル）が再度出力される。しかしながら、割込み処理ルーチン K の復帰時には、アドレス一致割込み一時禁止レジスタ 5 8 のフラグデータ F D 1 は“0”から“1”に書き換えられているので（ステップ K 3）、アドレス一致割込み一時禁止レジスタ 5 8 からはハイレベル信号がインバータ 5 6 に対して出力され、このためインバータ 5 6 からはローレベル信号が A N D ゲート 5 4 に対して出力されるので、フリップフロップ 6 2 の出力端子 Q からアドレス一致割込み要求信号が出力されることはなく、かくして A N D ゲート 6 6 からアドレス一致割込み信号が割込み制御部 2 0 に出力されて割込み処理ルーチン K が繰り返し実行されるというような事態は回避される。

【0 1 1 4】

なお、割込み処理ルーチン K の復帰時、アドレス比較器 5 2 から出力したアドレス一致信号はリセット回路 6 0 に対して出力され、このときリセット回路 6 0 からリセット信号 R E 1 がアドレス一致割込み一時禁止レジスタ 5 8 に出力されて、そのフラグデータ F D 1 が“1”から“0”に初期化され、これによりアドレス一致割込み一時禁止レジスタ 5 8 からインバータ 5 6 への出力信号も初期状態のローレベルに戻される。

【0 1 1 5】

更に、上述した第 3 のプログラム修正例（変更プログラム）に従う C P U 1 0 の動作について以下に説明する。

【0 1 1 6】

上述の第 1 及び第 2 のプログラム修正例と同様に、プログラムカウンタ 2 6 の更新アドレスが処理ルーチン E のコール命令の格納されたアドレス（4 バイト）

の最初の一バイト分のアドレス9820H に到達すると、その更新アドレス9820H は回路素子48₂ の比較アドレスレジスタ50に格納された比較アドレスデータ[9820H] と一致することとなる。双方のアドレスデータが互いに一致すると、回路素子48₂ のアドレス比較器52からはアドレス一致信号（ハイレベル）が出力され、これによりフリップフロップ62の出力端子Qからはアドレス一致割込み要求信号（ハイレベル）が出力され、このとき回路素子48₂ のアドレス一致割込み許可レジスタのフラグデータFE2 には“1”が与えられているので、回路素子48₂ のANDゲート66からはアドレス一致割込み信号が割込み制御部40 に対して出力される。

【0117】

回路素子48₂ のANDゲート66から出力されたアドレス一致割込み信号が割込み制御部40に出力されると、制御ユニット20では、処理ルーチンEのコール命令が格納されたアドレス(9820H~9823H)にアクセスすることなく割込み移行処理が上述したような割込み移行シーケンス（図21）に従って実行される。即ち、プログラムカウンタ26に保持されているアドレスデータ9820H がRAM 14のスタックメモリ領域に一時的に退避されると共にその時点での状態レジスタ38のステータス情報（種々のフラグデータ）もRAM 14のスタックメモリ領域に一時的に退避させられる。次いで、プログラムカウンタ26には回路素子48₂ のベクタ・アドレス・レジスタ70内のベクタ・アドレス・データ0A00H が設定されて、RAM 14の割込み処理ルーチンLの先頭アドレス0A00H に分岐し、割込み処理ルーチンLが実行される。勿論、上述の場合と同様に、割込み移行処理時には、状態レジスタ38の割込み許可フラグが“0”とされ、これにより緊急時の割込み処理（NMI）及びソフトウェア割込み処理以外の全割込み処理が禁止される。

【0118】

割込み処理ルーチンLでは、まず、ステップL1で回路素子48₂ のアドレス一致割込み要求クリアレジスタ74のフラグデータFC2 が“0”から“1”に書き換えられ、これによりクリア信号（ハイレベル）がアドレス一致割込み要求クリアレジスタ74からリセット回路72に対して出力される。上述の場合と同様

に、このクリア信号の出力により、これにより回路素子 48_2 のフリップフロップ 62 のラッチ状態が解除されて、その出力端子Qからのアドレス一致割込み要求信号の出力が停止され、またアドレス一致割込み要求クリアレジスタのフラグデータFC2が“1”から“0”に書き戻される。

【0119】

ステップL2では、ROM12上の処理ルーチンGのコール命令が格納されたアドレス(9830H～9833H)にアクセスされ、処理ルーチンG(先頭アドレス7000H)がコールされて実行される。要するに、処理ルーチンEを実行するステップ2808の直前で処理ルーチンGが実行されることになる。処理ルーチンGの実行が完了すると、ステップL3に進み、そこでアドレス一致割込み一時禁止レジスタ58のフラグデータFD2が“0”から“1”に書き換えられる。

【0120】

その後、変更プログラムでの最後の割込み処理からの復帰(リターン)命令により、割込み復帰処理が上述したような割込み復帰シーケンス(図24)に従って実行される。即ち、RAM14のスタックメモリ領域に一時的に退避されていたステータス情報が状態レジスタ38に戻され、またRAM14のスタックメモリ領域に一時的に退避されていた割込み時のアドレスデータ9820Hがプログラムカウンタ26に再び設定される。

【0121】

ところで、割込み処理ルーチンKの場合と同様に、割込み処理ルーチンLの復帰時、プログラムカウンタ26には上述したようにRAM14のスタックメモリ領域に一時的に退避されていた割込み発生時のアドレスデータ9820Hが設定されるために、そのアドレスデータ9820Hは回路素子 48_2 の比較アドレスレジスタ50内の比較アドレスデータと再び一致することとなり、このため回路素子 48_2 のアドレス比較器52からはアドレス一致信号(ハイレベル)が再度出力される。しかしながら、割込み処理ルーチンKの復帰時の場合と同様に、アドレス一致割込み一時禁止レジスタ58のフラグデータFD2は“0”から“1”に書き換えられているので(ステップL3)、ANDゲート66からアドレス一致割込み信号が割込み制御部20に出力されて割込み処理ルーチンLが繰り返し実行され

るというような事態は回避される。

【0122】

なお、割込み処理ルーチンKの場合と同様、割込み処理ルーチンLの復帰時でも、アドレス比較器52から出力したアドレス一致信号はリセット回路60に対して出力され、このときリセット回路60からリセット信号RE1がアドレス一致割込み一時禁止レジスタ58に出力されて、そのフラグデータFD2が“1”から“0”に初期化され、これによりアドレス一致割込み一時禁止レジスタ58からインバータ56への出力信号も初期状態のローレベルに戻される。

【0123】

以上の記載から明らかなように、本発明によれば、ROM12のプログラムに対して任意の箇所に新たな追加プログラムを追加することも、またROM12のプログラムのうちの一連の命令群（サブルーチン）を該プログラムの他の任意の箇所で実行させることもできる。

【0124】

なお、割込み処理ルーチンK及びLのそれぞれの割込み復帰時、必ずしも割込みが発生したアドレスに復帰することなく、割込み処理ルーチンJの場合と同様に任意のアドレスに分岐してもよい。勿論、そのような場合には、割込み処理ルーチンK及びLの最後に、分岐されるべきアドレスへのジャンプ命令が書き込まれ、このときは割込み復帰処理24は実行されない。

【0125】

ところで、先にも述べたように、アドレス一致割込み監視レジスタ64はROM12のプログラムの修正には直接的には関与しないが、しかし例えばROM12のプログラムの中でデバッキング等に利用することが可能である。例えば、ROM12のプログラムが正常に機能しないとき、該プログラムの実行中にアドレス一致割込み監視レジスタ64を用いて所定のアドレスにアクセスしたか否かを判別することが可能である。詳述すると、ROM12のプログラムを実行させる前に回路素子48_nのそれぞれの比較アドレスレジスタ50に該プログラムの実行中に本来アクセスされるべき所定アドレスが書き込まれ、もしそれら所定アドレスの各々にアクセスされれば、その対応アドレス一致割込み監視レジスタ64の

フラグデータ(FM0、FM1、…FM6、FM7)は“0”から“1”に書き換えられることになるので、そのフラグデータの値を監視することにより、ROM12のプログラムの実行中に所定アドレスにアクセスしたか否かが確認され得る。かくして、ROM12のプログラムが正常に機能しないとき、そのプログラムのどの箇所にバグがあるかを速やかに突き止めることができる。

【0126】

なお、以上述べたようなデバッキング時には、回路素子48_nのアドレス一致割込み許可レジスタ68のすべてのフラグデータ(FE0、FE1、…FE6及びFE7)には“0”が設定され、ANDゲート66からアドレス一致割込み信号が出力されないようにされることは勿論である。

【0127】

図32を参照すると、アドレス一致割込み制御部48の第1の実施形態の変形実施形態が示され、同図では、図2に示した第1の実施形態と同様な構成要素については同じ参照符号で示されている。この変形実施形態にあつては、各回路素子48_nのアドレス比較器52はプログラムカウンタ26に接続されずにアドレスラッチ76を介してプログラムカウンタ26のアドレスバスに接続される。

【0128】

図32の変形実施形態では、プログラムカウンタ26で逐次更新されるアドレスはそのアドレスバスから読み取ることが可能であり、プログラムカウンタ26で逐次更新されるアドレスはアドレスバスを介してアドレスラッチ76でアドレスデータとして順次取り込まれ、そのアドレスデータが8つの回路素子48_nのそれぞれのアドレス比較器52で比較アドレスレジスタ50の比較アドレスデータ(CA0、CA1、…CA6及びCA7)のいずれかと一致するか否か比較される。アドレスラッチ68のアドレスデータが比較アドレスレジスタ50のいずれかのアドレスデータと一致したとき、アドレス比較器52からアドレス一致信号(ハイレベル)が出力される。要するに、図32の変形実施形態でも、図2の第1の実施形態と同様な態様でROM12のメインプログラムに対して修正を加えることが可能である。

【0129】

図 3 3 を参照すると、アドレス一致割込み制御部 4 8 の第 2 の実施形態を構成する回路素子がブロック図として示され、この第 2 の実施形態でも、アドレス一致割込み制御部 4 8 (図 1) には図 3 3 に示すような回路素子が 8 つ設けられ、それら 8 個の回路素子が参照符号 80_n ($n = 0, 1, \dots, 6, 7$) で代表的に示されている。図 3 3 から明らかなように、図 2 に示す構成要素と同様な構成要素については同じ参照符号が用いられ、それら構成要素の個々の機能も第 1 の実施形態の場合と実質的に同じである。なお、第 1 の実施形態の場合と同様に、ROM 1 2 内のプログラムに修正を施し得る箇所は回路素子 80_n の個数に一致するので、ROM 1 2 のプログラムの任意の 8 箇所に修正を施すことが可能となる。

【0 1 3 0】

第 1 の実施形態では、ベクタ・アドレス・レジスタ 7 0 はバスを介してプログラムカウンタ 2 6 に接続されているのに対して、第 2 の実施形態では、ベクタ・アドレス・レジスタ 7 0 はバスを介してベクタ・アドレス・テンポラリ・レジスタ 7 8 に接続される。この点で第 2 の実施形態は第 1 の実施形態とは構成的に異なるだけであり、その他の点については双方の実施形態は実質的に同じである。ベクタ・アドレス・テンポラリ・レジスタ 7 8 は 16 ビット構成のレジスタとされ、そのビット数は ROM 1 2 のアドレスのビット数と同じである。また、ベクタ・アドレス・レジスタ 7 0 は制御バス、アドレスバス及びデータバスを介してマイクロコンピュータの各部に適宜接続され、CPU 1 0 の立上げ時にリセット信号によって初期化されるようになっている。

【0 1 3 1】

第 1 の実施形態と異なる機能的な相違点としては、第 2 の実施形態では、各回路素子 80_n のそれぞれの AND ゲート 6 6 からアドレス一致割込み信号が割込み制御部 4 0 に対して出力されると、CPU 1 0 の制御ユニット 2 0 では、割込み移行処理が図 3 4 に示すような割込み移行シーケンスに従って実行される点が挙げられ、またその他の相違点としては、ROM 1 2 の領域には図 3 5 に示すように所定のアドレス例えば 8000H 以降に割込み処理ルーチン M が予め格納されている点も挙げられる。

【0 1 3 2】

以上の点を除けば、第2の実施形態は第1の実施形態と実質的に同じであり、第1の実施形態で説明した第1、第2及び第3のプログラム修正例に倣って、第2の実施形態によるCPU10の動作について以下に説明する。

【0133】

まず、第2の実施形態での第1のプログラム修正例（修正プログラム）に従うCPU10の動作について以下に説明する。

【0134】

CPU10の動作中、プログラムカウンタ26でアドレスが逐次更新され、その更新アドレスが処理ルーチンBのコール命令の格納されたアドレス（4バイト）の最初の一バイト分のアドレス9808Hに到達すると、その更新アドレス9808Hは回路素子48₀の比較アドレスレジスタ50に格納された比較アドレスデータ[9808H]と一致することとなる。双方のアドレスデータが互いに一致すると、第1の実施形態の場合と同様な態様で、回路素子80₀のANDゲート66からはアドレス一致割込み信号が割込み制御部40に対して出力される。

【0135】

回路素子80₀のANDゲート66から出力されたアドレス一致割込み信号が割込み制御部40に出力されると、制御ユニット20では、処理ルーチンBのコール命令が格納されたアドレス(9808H～980BH)にアクセスすることなく割込み移行処理が図34に示すような割込み移行シーケンスに従って実行される。

【0136】

まず、ステップ3301では、プログラムカウンタ26に保持されているアドレスデータ9808HがRAM14のスタックメモリ領域に一時的に退避される。次いで、ステップ3302では、状態レジスタ38のステータス情報（種々のフラグデータ）もRAM14のスタックメモリ領域に一時的に退避させられる。続いて、ステップ3303では、状態レジスタ38に割り当てられた割込み許可フラグが“0”とされ、これにより緊急時の割込み処理（NMI）及びプログラムによる割込み処理以外の全ての割込み処理が禁止される。なお、先にも述べたように、以上の処理は従来のマイクロコンピュータで実施される通常の割込み処理に含まれるものである。

【0137】

ステップ3304では、回路素子80₀のベクタ・アドレス・レジスタ70からベクタ・アドレス・データ0600H (VA0) がベクタ・アドレス・テンポラリ・レジスタ78に設定される。なお、8つの回路素子80_nのベクタ・アドレス・レジスタ50のうちのどのベクタ・アドレス・データ(VA0、VA1、…VA6、VA7)をベクタ・アドレス・テンポラリ・レジスタ78に設定するかについては、8つの回路素子80_nのうちのどのベクタ・アドレス・レジスタ50にアドレス一致割込み信号が入力されているかで決定され、この第1のプログラム修正例では、回路素子80₀のベクタ・アドレス・レジスタ50にアドレス一致割込み信号が入力されることになるので、そのベクタ・アドレス・データ0600H (VA0) がベクタ・アドレス・テンポラリ・レジスタ78に設定される。

【0138】

ステップ3305では、ROM12上に存在する種々の割込みベクタ領域からアドレス一致割込みに対応するベクタアドレスがベクタ・アドレス・データとしてプログラムカウンタ26に設定される。即ち、本実施形態では、プログラムカウンタ26には割込み処理ルーチンMの先頭アドレス8000H (図35) が設定される。次いでステップ3306ではプログラムカウンタ26で示される割込み処理ルーチンMに分岐する。

【0139】

割込み処理ルーチンMが実行されると、CPU10の制御ユニット20からはベクタ・アドレス・テンポラリ・レジスタ78に設定されたベクタアドレス0600Hに分岐すべき命令が発せられ、これによりプログラムカウンタ26のアドレスはベクタアドレス0600Hによって更新され、かくして第1の実施形態の場合と同様に、RAM14の領域の先頭アドレス0600Hに分岐して、割込み処理ルーチンJが実行されることになる。なお、割込み処理ルーチンJでの修正ルーチンの完了後の割込み処理からの復帰は第1の実施形態の場合と同様である。

【0140】

次に、第2の実施形態での第2のプログラム修正例(追加プログラム)に従うCPU10の動作について以下に説明する。

【0141】

CPU10の動作中、プログラムカウンタ26でアドレスが逐次更新され、その更新アドレスが処理ルーチンDのコール命令の格納されたアドレス（4バイト）の最初の一バイト分のアドレス9818Hに到達すると、その更新アドレスデータ9818Hは回路素子80₁の比較アドレスレジスタ50に格納された比較アドレスデータ[9818H]と一致することとなる。双方のアドレスデータが互いに一致すると、第1の実施形態の場合と同様な態様で、回路素子80₁のANDゲート66からはアドレス一致割込み信号が割込み制御部40に対して出力される。

【0142】

回路素子80₁のANDゲート66から出力されたアドレス一致割込み信号が割込み制御部40に出力されると、制御ユニット20では、処理ルーチンDのコール命令が格納されたアドレス(9818H~981BH)にアクセスすることなく割込み移行処理が図34に示すような割込み移行シーケンスに従って実行される。なお、ステップ3301ないし3304では上述の場合と同様な処理が行われる（勿論、このときRAM14のスタックメモリ領域に一時的に退避されるアドレスデータは9818Hとなる）。

【0143】

ステップ3304では、回路素子80₁のベクタ・アドレス・レジスタ70からベクタ・アドレス・データ0800H (VA1)がベクタ・アドレス・テンポラリ・レジスタ78に設定される。なお、上述の場合と同様に、8つの回路素子80_nのベクタ・アドレス・レジスタ50のうちどのベクタ・アドレス・データ(VA0、VA1、…VA6、VA7)をベクタ・アドレス・テンポラリ・レジスタ78に設定するかについては、8つの回路素子80_nのうちどのベクタ・アドレス・レジスタ50にアドレス一致割込み信号が入力されているかで決定され、この第2のプログラム修正例では、回路素子80₁のベクタ・アドレス・レジスタ50にアドレス一致割込み信号が入力されることになるので、そのベクタ・アドレス・データ0800H (VA1)がベクタ・アドレス・テンポラリ・レジスタ78に設定される。

【0144】

ステップ3305では、プログラムカウンタ26に上述の場合と同様にアドレ

ス一致割込みに対応するベクタアドレスとして割込み処理ルーチンMの先頭アドレス8000H が設定され、次いでステップ3306ではプログラムカウンタ26で示される割込み処理ルーチンMに分岐する。

【0145】

割込み処理ルーチンMが実行されると、CPU10の制御ユニット20からはベクタ・アドレス・テンポラリ・レジスタ78に設定されたベクタアドレス0800H に分岐すべき命令が発せられ、これによりプログラムカウンタ26のアドレスはベクタアドレス0800H によって更新され、かくして第1の実施形態の場合と同様に、RAM14の領域の先頭アドレス0800H に分岐して、割込み処理ルーチンKが実行されることになる。なお、割込み処理ルーチンKの完了後、図24の割込み復帰シーケンスに従って復帰処理が行われることは第1の実施形態の場合と同様である。

【0146】

続いて、第2の実施形態での第3のプログラム修正例（変更プログラム）に従うCPU10の動作について以下に説明する。

【0147】

CPU10の動作中、プログラムカウンタ26でアドレスが逐次更新され、その更新アドレスが処理ルーチンEのコール命令の格納されたアドレス（4バイト）の最初の一バイト分のアドレス9820H に到達すると、その更新アドレスデータ9820H は回路素子80₂の比較アドレスレジスタ50に格納された比較アドレスデータ[9820H]と一致することとなる。双方のアドレスデータが互いに一致すると、第1の実施形態の場合と同様な態様で、回路素子80₂のANDゲート66からはアドレス一致割込み信号が割込み制御部40に対して出力される。

【0148】

回路素子80₂のANDゲート66から出力されたアドレス一致割込み信号が割込み制御部40に出力されると、制御ユニット20では、処理ルーチンEのコール命令が格納されたアドレス(9820H~9823H)にアクセスすることなく割込み移行処理が図34に示すような割込み移行シーケンスに従って実行される。なお、ステップ3301ないし3304では上述の場合と同様な処理が行われる（勿論

、このときRAM 14のスタックメモリ領域に一時的に退避されるアドレスデータは9820H となる)。

【0149】

ステップ3304では、回路素子80₂のベクタ・アドレス・レジスタ70からベクタ・アドレス・データ0A00H (VA2) がベクタ・アドレス・テンポラリ・レジスタ78に設定される。なお、上述の場合と同様に、8つの回路素子80_nのベクタ・アドレス・レジスタ50のうちのどのベクタ・アドレス・データ(VA0、VA1、…VA6、VA7)をベクタ・アドレス・テンポラリ・レジスタ78に設定するかについては、8つの回路素子80_nのうちのどのベクタ・アドレス・レジスタ50にアドレス一致割込み信号が入力されているかで決定され、この第3のプログラム修正例では、回路素子80₂のベクタ・アドレス・レジスタ50にアドレス一致割込み信号が入力されることになるので、そのベクタ・アドレス・データ0A00H (VA2) がベクタ・アドレス・テンポラリ・レジスタ78に設定される。

【0150】

ステップ3305では、プログラムカウンタ26に上述の場合と同様にアドレス一致割込みに対応するベクタアドレスとして割込み処理ルーチンMの先頭アドレス8000H が設定され、次いでステップ3306ではプログラムカウンタ26で示される割込み処理ルーチンMに分岐する。

【0151】

割込み処理ルーチンMが実行されると、CPU 10の制御ユニット20からはベクタ・アドレス・テンポラリ・レジスタ78に設定されたベクタアドレス0A00H に分岐すべき命令が発せられ、これによりプログラムカウンタ26のアドレスはベクタアドレス0A00H によって更新され、かくして第1の実施形態の場合と同様に、RAM 14の領域の先頭アドレス0A00H に分岐して、割込み処理ルーチンLが実行されることになる。なお、割込み処理ルーチンLの完了後、図24の割込み復帰シーケンスに従って復帰処理が行われることは第1の実施形態の場合と同様である。

【0152】

以上で述べた第2の実施形態において、割込み処理ルーチンM (図35) に8

つの回路素子 80_n のアドレス一致割込み要求クリアレジスタ 74 のフラグデータ FC0、FC1、…FC6 及び FC7 のすべてを強制的に“0”にするステップを設けてもよく、この場合には割込み処理ルーチン J、K 及び L (30) のそれぞれからステップ J1、K1 及び L1 を省くことが可能である。

【0153】

図 36 を参照すると、アドレス一致割込み制御部 48 の第 2 の実施形態 (図 33) の変形実施形態が示され、同図では、図 33 に示した第 2 の実施形態と同様な構成要素については同じ参照符号で示されている。この変形実施形態では、アドレス比較器 52 はプログラムカウンタ 26 に接続されずにアドレスラッチ 82 を介してプログラムカウンタ 26 のアドレスバスに接続され、この構成は第 1 の実施形態 (図 2) の変形実施形態として図 32 に示したものと同様である。

【0154】

図 36 の変形実施形態では、プログラムカウンタ 26 で逐次更新されるアドレスはアドレスラッチ 82 にアドレスデータとして順次取り込まれ、そのアドレスデータが 8 つの回路素子 80_n のそれぞれのアドレス比較器 52 で比較アドレスレジスタ 50 の比較アドレスデータ (CA0、CA1、…CA6 及び CA7) のいずれかと一致するか否か比較される。アドレスラッチ 82 のアドレスデータが比較アドレスレジスタ 50 のいずれかのアドレスデータと一致したとき、アドレス比較器 52 からアドレス一致信号 (ハイレベル) が出力される。要するに、図 36 の変形実施形態でも、図 33 の第 2 の実施形態と同様な態様で ROM 12 のメインプログラムに対して修正を加えることが可能である。

【0155】

【発明の効果】

以上の記載から明らかなように、本発明によるプログラム修正機能内蔵マイクロコンピュータにおいては、修正機能が不要な場合には、その処理時間に実質的な影響を与えることなく、読出し専用メモリ (ROM) のプログラムの全ての任意の箇所に対して修正を必要に応じて加えることができるので、マイクロコンピュータのプログラム設計に多大な融通性を与えるだけでなく、その製造コストの改善に大きく寄与し得る。

【図面の簡単な説明】

【図 1】

本発明によるマイクロコンピュータの概略ブロック図である。

【図 2】

図 1 に示すマイクロコンピュータの一構成要素であって、本発明に従って構成されたアドレス一致割込み制御部の第 1 の実施形態を示すブロック図である。

【図 3】

図 1 に示すマイクロコンピュータの一構成要素として設けられる E E P R O M のメモリ領域を概念的に示す模式図である。

【図 4】

図 1 のアドレス一致割込み制御部を構成する第 1 の回路素子の比較アドレスレジスタ、アドレス一致割込み許可レジスタ及びベクタ・アドレス・レジスタに書き込まれたデータを概念的に示す模式図である。

【図 5】

図 1 のアドレス一致割込み制御部を構成する第 2 の回路素子の比較アドレスレジスタ、アドレス一致割込み許可レジスタ及びベクタ・アドレス・レジスタに書き込まれたデータを概念的に示す模式図である。

【図 6】

図 1 のアドレス一致割込み制御部を構成する第 3 の回路素子の比較アドレスレジスタ、アドレス一致割込み許可レジスタ及びベクタ・アドレス・レジスタに書き込まれたデータを概念的に示す模式図である。

【図 7】

図 1 のアドレス一致割込み制御部を構成する第 4 の回路素子の比較アドレスレジスタ、アドレス一致割込み許可レジスタ及びベクタ・アドレス・レジスタに書き込まれたデータを概念的に示す模式図である。

【図 8】

図 1 のアドレス一致割込み制御部を構成する第 5 の回路素子の比較アドレスレジスタ、アドレス一致割込み許可レジスタ及びベクタ・アドレス・レジスタに書き込まれたデータを概念的に示す模式図である。

【図 9】

図 1 のアドレス一致割込み制御部を構成する第 6 の回路素子の比較アドレスレジスタ、アドレス一致割込み許可レジスタ及びベクタ・アドレス・レジスタに書き込まれたデータを概念的に示す模式図である。

【図 1 0】

図 1 のアドレス一致割込み制御部を構成する第 7 の回路素子の比較アドレスレジスタ、アドレス一致割込み許可レジスタ及びベクタ・アドレス・レジスタに書き込まれたデータを概念的に示す模式図である。

【図 1 1】

図 1 のアドレス一致割込み制御部を構成する第 8 の回路素子の比較アドレスレジスタ、アドレス一致割込み許可レジスタ及びベクタ・アドレス・レジスタに書き込まれたデータを概念的に示す模式図である。

【図 1 2】

図 1 に示すマイクロコンピュータの一構成要素として設けられる RAM のメモリ領域を概念的に示す模式図である。

【図 1 3】

図 1 のアドレス一致割込み制御部を構成する第 1 の回路素子のアドレス一致割込み一時禁止レジスタ、アドレス一致割込み要求クリアレジスタ及びアドレス一致割込み監視レジスタに書き込まれたデータを概念的に示す模式図である。

【図 1 4】

図 1 のアドレス一致割込み制御部を構成する第 2 の回路素子のアドレス一致割込み一時禁止レジスタ、アドレス一致割込み要求クリアレジスタ及びアドレス一致割込み監視レジスタに書き込まれたデータを概念的に示す模式図である。

【図 1 5】

図 1 のアドレス一致割込み制御部を構成する第 3 の回路素子のアドレス一致割込み一時禁止レジスタ、アドレス一致割込み要求クリアレジスタ及びアドレス一致割込み監視レジスタに書き込まれたデータを概念的に示す模式図である。

【図 1 6】

図 1 のアドレス一致割込み制御部を構成する第 4 の回路素子のアドレス一致割

込み一時禁止レジスタ、アドレス一致割込み要求クリアレジスタ及びアドレス一致割込み監視レジスタに書き込まれたデータを概念的に示す模式図である。

【図 1 7】

図 1 のアドレス一致割込み制御部を構成する第 5 の回路素子のアドレス一致割込み一時禁止レジスタ、アドレス一致割込み要求クリアレジスタ及びアドレス一致割込み監視レジスタに書き込まれたデータを概念的に示す模式図である。

【図 1 8】

図 1 のアドレス一致割込み制御部を構成する第 6 の回路素子のアドレス一致割込み一時禁止レジスタ、アドレス一致割込み要求クリアレジスタ及びアドレス一致割込み監視レジスタに書き込まれたデータを概念的に示す模式図である。

【図 1 9】

図 1 のアドレス一致割込み制御部を構成する第 7 の回路素子のアドレス一致割込み一時禁止レジスタ、アドレス一致割込み要求クリアレジスタ及びアドレス一致割込み監視レジスタに書き込まれたデータを概念的に示す模式図である。

【図 2 0】

図 1 のアドレス一致割込み制御部を構成する第 8 の回路素子のアドレス一致割込み一時禁止レジスタ、アドレス一致割込み要求クリアレジスタ及びアドレス一致割込み監視レジスタに書き込まれたデータを概念的に示す模式図である。

【図 2 1】

図 2 のアドレス一致割込み制御部からのアドレス一致割込み信号の出力時に図 1 のマイクロコンピュータで実行される割込み移行処理を示すフローチャートである。

【図 2 2】

図 2 に示すアドレス一致割込み制御部のフリップフロップのラッチ状態を解除するためのリセット回路の詳細図である。

【図 2 3】

図 2 2 のリセット回路の動作を示すタイミングチャートである。

【図 2 4】

図 2 のアドレス一致割込み制御部からのアドレス一致割込み信号の出力により

実行された割込み処理の完了後に図 1 のマイクロコンピュータで実行される割込復帰処理を示すフローチャートである。

【図 25】

図 2 に示すアドレス一致割込み制御部のアドレス一致割込み一時禁止レジスタを初期化するためのリセット回路の詳細図である。

【図 26】

図 22 のリセット回路の動作を示すタイミングチャートである。

【図 27】

図 1 に示すマイクロコンピュータを所定の電子機器に搭載した際に該マイクロコンピュータによって実行されるメインルーチンを例示的に示すフローチャートである。

【図 28】

図 27 のメインルーチンで実行される例示的プログラムの一部を示すフローチャートである。

【図 29】

図 28 に示した例示的プログラムの各ステップ処理を図 1 のマイクロコンピュータのプログラムカウンタで逐次更新されるアドレスとの関係を概念的に示す模式図である。

【図 30】

図 28 に示された例示的プログラムからコールされる種々のサブルーチンのフローチャートを図 1 のマイクロコンピュータの ROM の所定アドレスに格納された状態で概念的に示す模式図である。

【図 31】

図 1 のマイクロコンピュータの ROM のプログラムに対する修正プログラムのフローチャートを該マイクロコンピュータの RAM に割込み処理ルーチンとして所定アドレスに格納された状態で概念的に示す模式図である。

【図 32】

図 2 に示すアドレス一致割込み制御部の第 1 の実施形態の変形実施形態を示すブロック図である。

【図 3 3】

図 1 に示すマイクロコンピュータの一構成要素であって、本発明に従って構成されたアドレス一致割込み制御部の第 2 の実施形態を示すブロック図である。

【図 3 4】

図 3 3 のアドレス一致割込み制御部からのアドレス一致割込み信号の出力時に図 1 のマイクロコンピュータで実行される割込み移行処理を示すフローチャートである。

【図 3 5】

図 3 0 に示す模式図と同様な模式図であって、図 3 3 のアドレス一致割込み制御部からのアドレス一致割込み信号の出力により図 1 のマイクロコンピュータで実行される割込み移行処理後に実行される割込み処理ルーチンのフローチャートを該マイクロコンピュータの ROM の所定アドレスに格納された状態で概念的に示す模式図である。

【図 3 6】

図 3 3 に示すアドレス一致割込み制御部の第 2 の実施形態の変形実施形態を示すブロック図である。

【符号の説明】

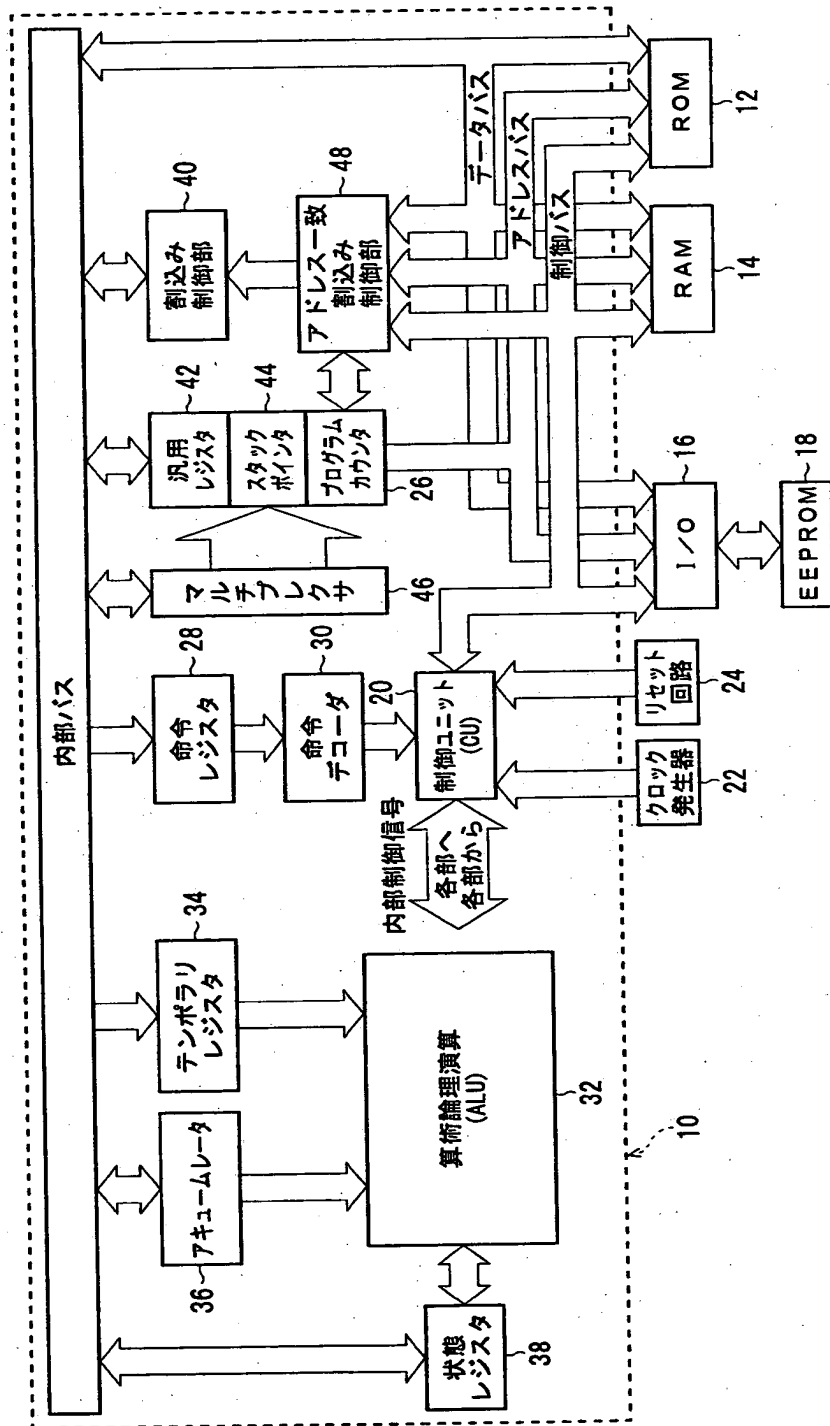
- 1 0 中央演算処理ユニット (CPU)
- 1 2 読出し専用メモリ (ROM)
- 1 4 書込み／読出し自在なメモリ (RAM)
- 1 6 入出力インターフェース (I/O)
- 1 8 電氣的に消去可能でしかもプログラム可能な読出し専用メモリ (EEPROM)
- 2 0 制御ユニット (CU)
- 2 6 プログラムカウンタ
- 2 8 命令レジスタ
- 3 0 命令デコーダ
- 3 2 算術論理演算ユニット (ALU)
- 4 0 割込み制御部

- 4 4 スタックポインタ
- 4 8 アドレス一致割込み制御部
- 5 0 比較アドレスレジスタ
- 5 2 アドレス比較器
- 5 4 ANDゲート
- 5 6 インバータ
- 5 8 アドレス一致割込み一時禁止レジスタ
- 6 0 リセット回路
- 6 2 フリップフロップ
- 6 4 アドレス一致割込み監視レジスタ
- 6 6 ANDゲート
- 6 8 アドレス一致割込み許可レジスタ
- 7 0 ベクタ・アドレス・レジスタ
- 7 2 リセット回路
- 7 4 アドレス一致割込み要求クリアレジスタ

【書類名】

図面

【図 1】



【図 3】

EEPROM18							
CA0 ₁₅	CA0 ₁₄	CA0 ₁₃	CA0 ₁₂	CA0 ₁₁	CA0 ₁₀	CA0 ₀₉	CA0 ₀₈
CA0 ₀₇	CA0 ₀₆	CA0 ₀₅	CA0 ₀₄	CA0 ₀₃	CA0 ₀₂	CA0 ₀₁	CA0 ₀₀
CA1 ₁₅	CA1 ₁₄	CA1 ₁₃	CA1 ₁₂	CA1 ₁₁	CA1 ₁₀	CA1 ₀₉	CA1 ₀₈
CA1 ₀₇	CA1 ₀₆	CA1 ₀₅	CA1 ₀₄	CA1 ₀₃	CA1 ₀₂	CA1 ₀₁	CA1 ₀₀
CA6 ₁₅	CA6 ₁₄	CA6 ₁₃	CA6 ₁₂	CA6 ₁₁	CA6 ₁₀	CA6 ₀₉	CA6 ₀₈
CA6 ₀₇	CA6 ₀₆	CA6 ₀₅	CA6 ₀₄	CA6 ₀₃	CA6 ₀₂	CA6 ₀₁	CA6 ₀₀
CA7 ₁₅	CA7 ₁₄	CA7 ₁₃	CA7 ₁₂	CA7 ₁₁	CA7 ₁₀	CA7 ₀₉	CA7 ₀₈
CA7 ₀₇	CA7 ₀₆	CA7 ₀₅	CA7 ₀₄	CA7 ₀₃	CA7 ₀₂	CA7 ₀₁	CA7 ₀₀
FE7	FE6	FE5	FE4	FE3	FE2	FE1	FE0
VA0 ₁₅	VA0 ₁₄	VA0 ₁₃	VA0 ₁₂	VA0 ₁₁	VA0 ₁₀	VA0 ₀₉	VA0 ₀₈
VA0 ₀₇	VA0 ₀₆	VA0 ₀₅	VA0 ₀₄	VA0 ₀₃	VA0 ₀₂	VA0 ₀₁	VA0 ₀₀
VA1 ₁₅	VA1 ₁₄	VA1 ₁₃	VA1 ₁₂	VA1 ₁₁	VA1 ₁₀	VA1 ₀₉	VA1 ₀₈
VA1 ₀₇	VA1 ₀₆	VA1 ₀₅	VA1 ₀₄	VA1 ₀₃	VA1 ₀₂	VA1 ₀₁	VA1 ₀₀
VA6 ₁₅	VA6 ₁₄	VA6 ₁₃	VA6 ₁₂	VA6 ₁₁	VA6 ₁₀	VA6 ₀₉	VA6 ₀₈
VA6 ₀₇	VA6 ₀₆	VA6 ₀₅	VA6 ₀₄	VA6 ₀₃	VA6 ₀₂	VA6 ₀₁	VA6 ₀₀
VA7 ₁₅	VA7 ₁₄	VA7 ₁₃	VA7 ₁₂	VA7 ₁₁	VA7 ₁₀	VA7 ₀₉	VA7 ₀₈
VA7 ₀₇	VA7 ₀₆	VA7 ₀₅	VA7 ₀₄	VA7 ₀₃	VA7 ₀₂	VA7 ₀₁	VA7 ₀₀
修正/追加プログラム書込み領域							
データ書込み領域							
空き領域							

【図 4】

回路素子48₀

比較アドレスレジスタ50

CA0 ₁₅	CA0 ₁₄	CA0 ₁₃	CA0 ₁₂	CA0 ₁₁	CA0 ₁₀	CA0 ₀₉	CA0 ₀₈
CA0 ₀₇	CA0 ₀₆	CA0 ₀₅	CA0 ₀₄	CA0 ₀₃	CA0 ₀₂	CA0 ₀₁	CA0 ₀₀

アドレス一致割込み許可レジスタ68

FE0

ベクタ・アドレス・レジスタ70

VA0 ₁₅	VA0 ₁₄	VA0 ₁₃	VA0 ₁₂	VA0 ₁₁	VA0 ₁₀	VA0 ₀₉	VA0 ₀₈
VA0 ₀₇	VA0 ₀₆	VA0 ₀₅	VA0 ₀₄	VA0 ₀₃	VA0 ₀₂	VA0 ₀₁	VA0 ₀₀

【図 5】

回路素子48₁

比較アドレスレジスタ50

CA1 ₁₅	CA1 ₁₄	CA1 ₁₃	CA1 ₁₂	CA1 ₁₁	CA1 ₁₀	CA1 ₀₉	CA1 ₀₈
CA1 ₀₇	CA1 ₀₆	CA1 ₀₅	CA1 ₀₄	CA1 ₀₃	CA1 ₀₂	CA1 ₀₁	CA1 ₀₀

アドレス一致割込み許可レジスタ68

FE1

ベクタ・アドレス・レジスタ70

VA1 ₁₅	VA1 ₁₄	VA1 ₁₃	VA1 ₁₂	VA1 ₁₁	VA1 ₁₀	VA1 ₀₉	VA1 ₀₈
VA1 ₀₇	VA1 ₀₆	VA1 ₀₅	VA1 ₀₄	VA1 ₀₃	VA1 ₀₂	VA1 ₀₁	VA1 ₀₀

【図 6】

回路素子48₂

比較アドレスレジスタ50

CA2 ₁₅	CA2 ₁₄	CA2 ₁₃	CA2 ₁₂	CA2 ₁₁	CA2 ₁₀	CA2 ₀₉	CA2 ₀₈
CA2 ₀₇	CA2 ₀₆	CA2 ₀₅	CA2 ₀₄	CA2 ₀₃	CA2 ₀₂	CA2 ₀₁	CA2 ₀₀

アドレス一致割込み許可レジスタ68

FE2

ベクタ・アドレス・レジスタ70

VA2 ₁₅	VA2 ₁₄	VA2 ₁₃	VA2 ₁₂	VA2 ₁₁	VA2 ₁₀	VA2 ₀₉	VA2 ₀₈
VA2 ₀₇	VA2 ₀₆	VA2 ₀₅	VA2 ₀₄	VA2 ₀₃	VA2 ₀₂	VA2 ₀₁	VA2 ₀₀

【図 7】

回路素子48₃

比較アドレスレジスタ50

CA3 ₁₅	CA3 ₁₄	CA3 ₁₃	CA3 ₁₂	CA3 ₁₁	CA3 ₁₀	CA3 ₀₉	CA3 ₀₈
CA3 ₀₇	CA3 ₀₆	CA3 ₀₅	CA3 ₀₄	CA3 ₀₃	CA3 ₀₂	CA3 ₀₁	CA3 ₀₀

アドレス一致割込み許可レジスタ68

FE3

ベクタ・アドレス・レジスタ70

VA3 ₁₅	VA3 ₁₄	VA3 ₁₃	VA3 ₁₂	VA3 ₁₁	VA3 ₁₀	VA3 ₀₉	VA3 ₀₈
VA3 ₀₇	VA3 ₀₆	VA3 ₀₅	VA3 ₀₄	VA3 ₀₃	VA3 ₀₂	VA3 ₀₁	VA3 ₀₀

【図 8】

回路素子48₄

比較アドレスレジスタ50

CA4 ₁₅	CA4 ₁₄	CA4 ₁₃	CA4 ₁₂	CA4 ₁₁	CA4 ₁₀	CA4 ₀₉	CA4 ₀₈
CA4 ₀₇	CA4 ₀₆	CA4 ₀₅	CA4 ₀₄	CA4 ₀₃	CA4 ₀₂	CA4 ₀₁	CA4 ₀₀

アドレス一致割込み許可レジスタ68

FE4

ベクタ・アドレス・レジスタ70

VA4 ₁₅	VA4 ₁₄	VA4 ₁₃	VA4 ₁₂	VA4 ₁₁	VA4 ₁₀	VA4 ₀₉	VA4 ₀₈
VA4 ₀₇	VA4 ₀₆	VA4 ₀₅	VA4 ₀₄	VA4 ₀₃	VA4 ₀₂	VA4 ₀₁	VA4 ₀₀

【図 9】

回路素子48₅

比較アドレスレジスタ50

CA5 ₁₅	CA5 ₁₄	CA5 ₁₃	CA5 ₁₂	CA5 ₁₁	CA5 ₁₀	CA5 ₀₉	CA5 ₀₈
CA5 ₀₇	CA5 ₀₆	CA5 ₀₅	CA5 ₀₄	CA5 ₀₃	CA5 ₀₂	CA5 ₀₁	CA5 ₀₀

アドレス一致割込み許可レジスタ68

FE5

ベクタ・アドレス・レジスタ70

VA5 ₁₅	VA5 ₁₄	VA5 ₁₃	VA5 ₁₂	VA5 ₁₁	VA5 ₁₀	VA5 ₀₉	VA5 ₀₈
VA5 ₀₇	VA5 ₀₆	VA5 ₀₅	VA5 ₀₄	VA5 ₀₃	VA5 ₀₂	VA5 ₀₁	VA5 ₀₀

【図 1 0】

回路素子48₆

比較アドレスレジスタ50

CA6 ₁₅	CA6 ₁₄	CA6 ₁₃	CA6 ₁₂	CA6 ₁₁	CA6 ₁₀	CA6 ₀₉	CA6 ₀₈
CA6 ₀₇	CA6 ₀₆	CA6 ₀₅	CA6 ₀₄	CA6 ₀₃	CA6 ₀₂	CA6 ₀₁	CA6 ₀₀

アドレス一致割込み許可レジスタ68

FE6

ベクタ・アドレス・レジスタ70

VA6 ₁₅	VA6 ₁₄	VA6 ₁₃	VA6 ₁₂	VA6 ₁₁	VA6 ₁₀	VA6 ₀₉	VA6 ₀₈
VA6 ₀₇	VA6 ₀₆	VA6 ₀₅	VA6 ₀₄	VA6 ₀₃	VA6 ₀₂	VA6 ₀₁	VA6 ₀₀

【図 1 1】

回路素子48₇

比較アドレスレジスタ50

CA7 ₁₅	CA7 ₁₄	CA7 ₁₃	CA7 ₁₂	CA7 ₁₁	CA7 ₁₀	CA7 ₀₉	CA7 ₀₈
CA7 ₀₇	CA7 ₀₆	CA7 ₀₅	CA7 ₀₄	CA7 ₀₃	CA7 ₀₂	CA7 ₀₁	CA7 ₀₀

アドレス一致割込み許可レジスタ68

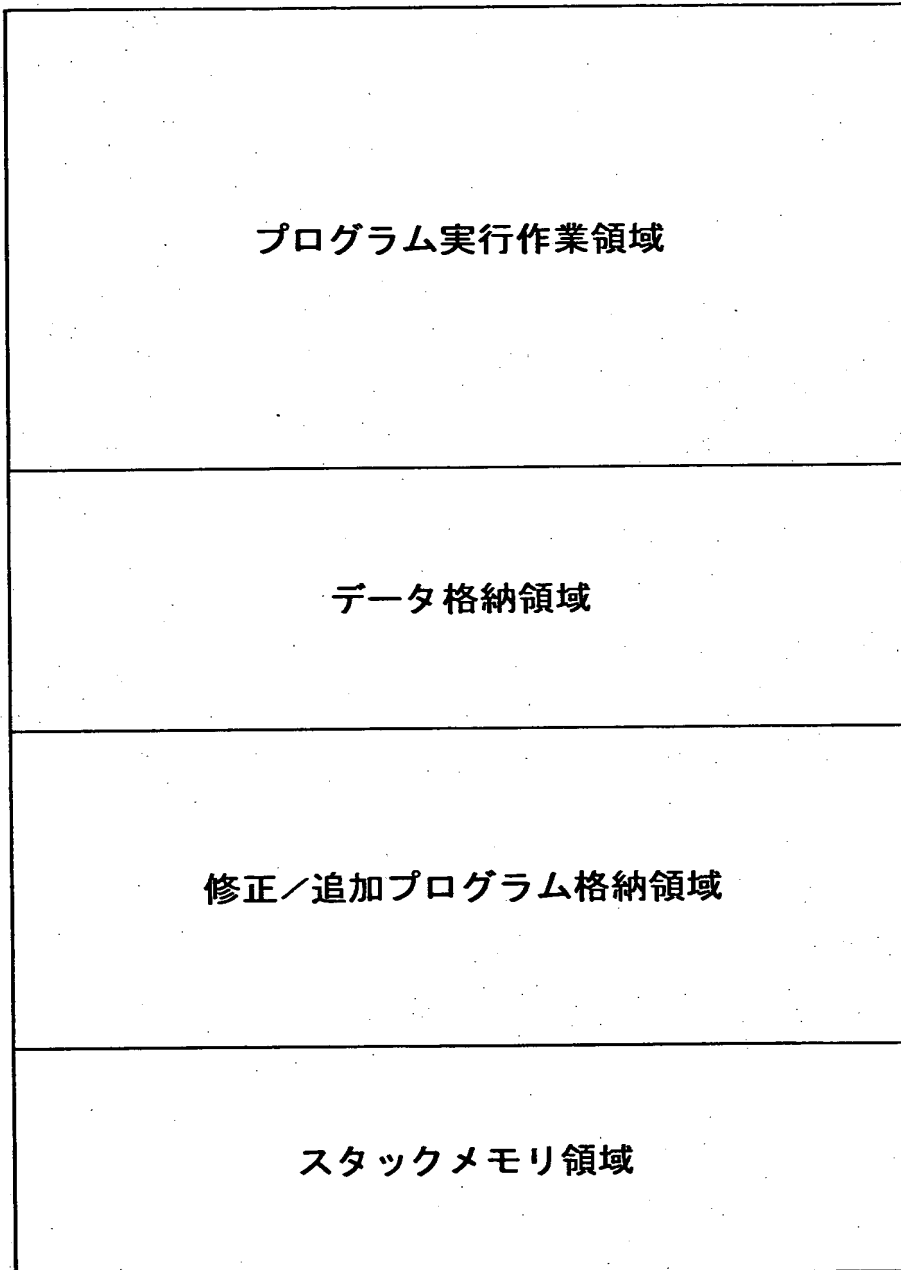
FE7

ベクタ・アドレス・レジスタ70

VA7 ₁₅	VA7 ₁₄	VA7 ₁₃	VA7 ₁₂	VA7 ₁₁	VA7 ₁₀	VA7 ₀₉	VA7 ₀₈
VA7 ₀₇	VA7 ₀₆	VA7 ₀₅	VA7 ₀₄	VA7 ₀₃	VA7 ₀₂	VA7 ₀₁	VA7 ₀₀

【図 1 2】

RAM14



【図 1 3】

回路素子48₀

アドレス一致割込み一時禁止レジスタ58

FD0

アドレス一致割込みクリアレジスタ74

FC0

アドレス一致割込み監視レジスタ64

FM0

【図 1 4】

回路素子48₁

アドレス一致割込み一時禁止レジスタ58

FD1

アドレス一致割込みクリアレジスタ74

FC1

アドレス一致割込み監視レジスタ64

FM1

【図 1 5】

回路素子48₂

アドレス一致割込み一時禁止レジスタ58

FD2

アドレス一致割込みクリアレジスタ74

FC2

アドレス一致割込み監視レジスタ64

FM2

【図 1 6】

回路素子48₃

アドレス一致割込み一時禁止レジスタ58

FD3

アドレス一致割込みクリアレジスタ74

FC3

アドレス一致割込み監視レジスタ64

FM3

【図17】

回路素子48₄

アドレス一致割込み一時禁止レジスタ58

FD4

アドレス一致割込みクリアレジスタ74

FC4

アドレス一致割込み監視レジスタ64

FM4

【図18】

回路素子48₅

アドレス一致割込み一時禁止レジスタ58

FD5

アドレス一致割込みクリアレジスタ74

FC5

アドレス一致割込み監視レジスタ64

FM5

【図 1 9】

回路素子48₆

アドレス一致割込み一時禁止レジスタ58

FD6

アドレス一致割込みクリアレジスタ74

FC6

アドレス一致割込み監視レジスタ64

FM6

【図 2 0】

回路素子48₇

アドレス一致割込み一時禁止レジスタ58

FD7

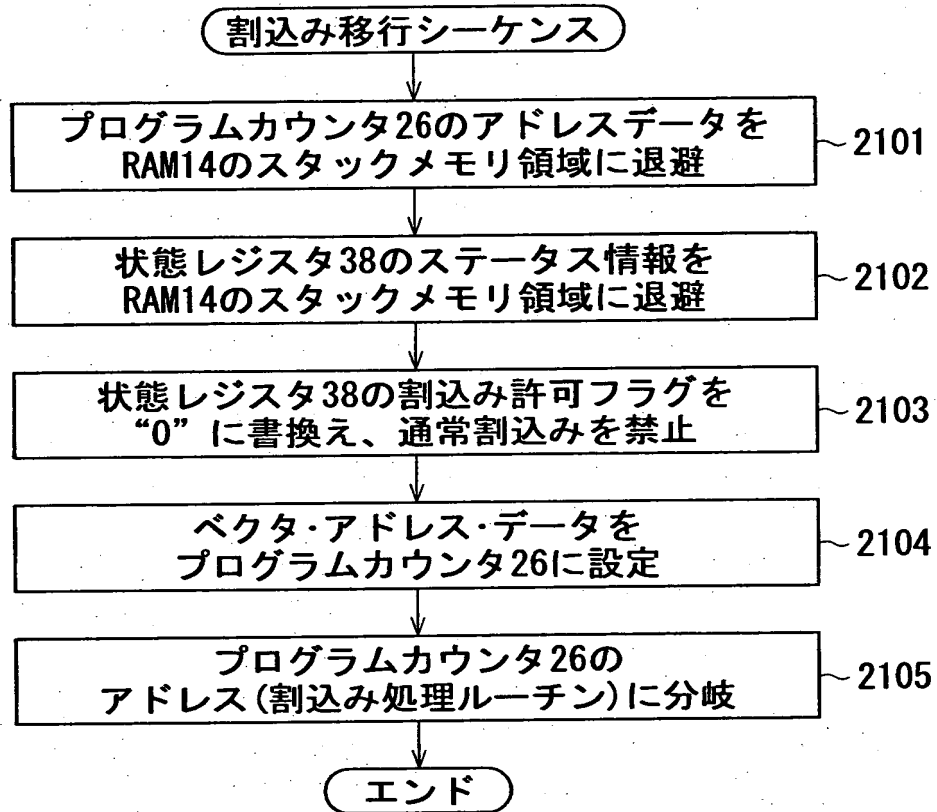
アドレス一致割込みクリアレジスタ74

FC7

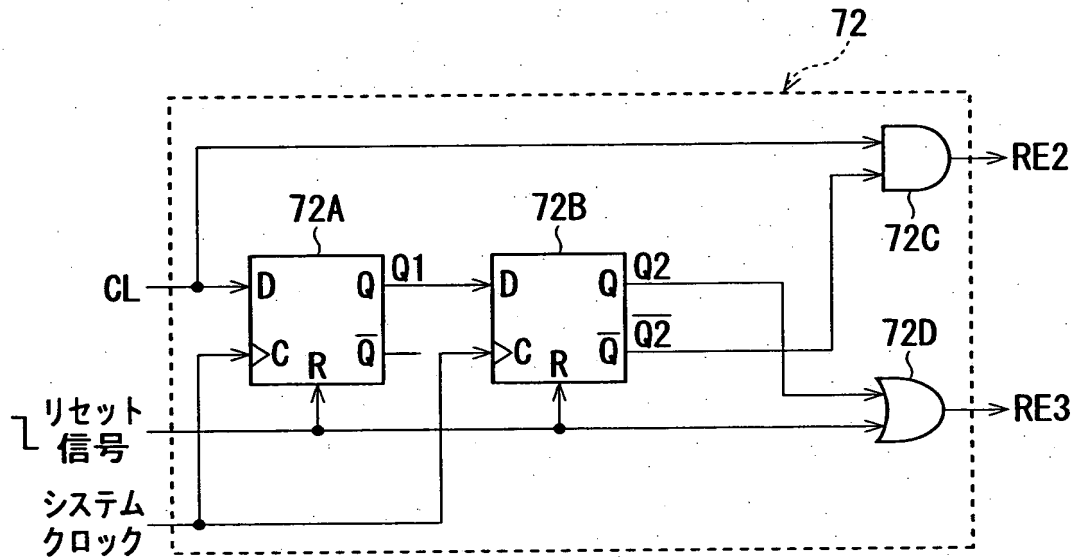
アドレス一致割込み監視レジスタ64

FM7

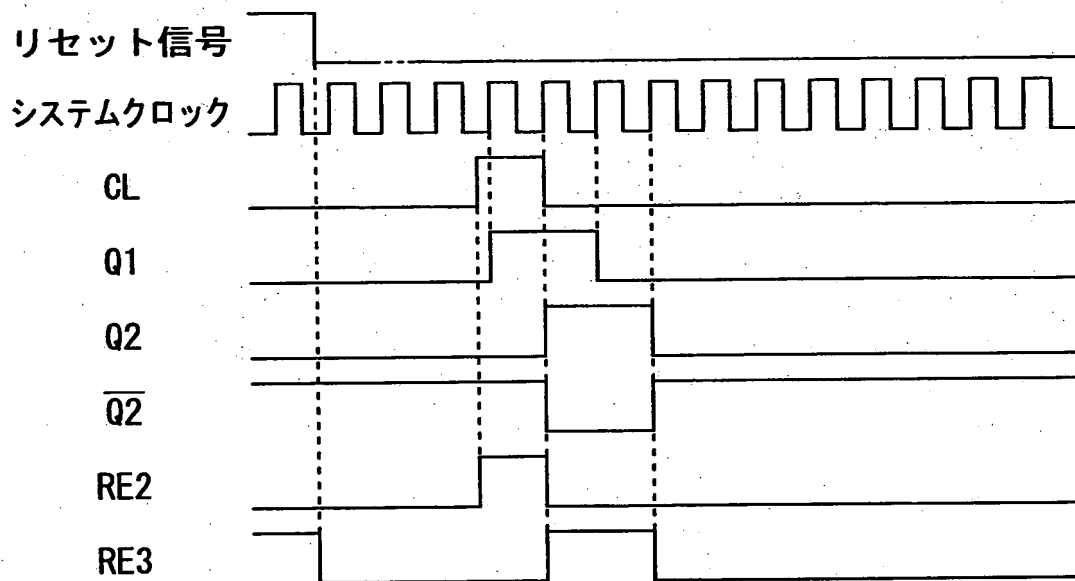
【図 2 1】



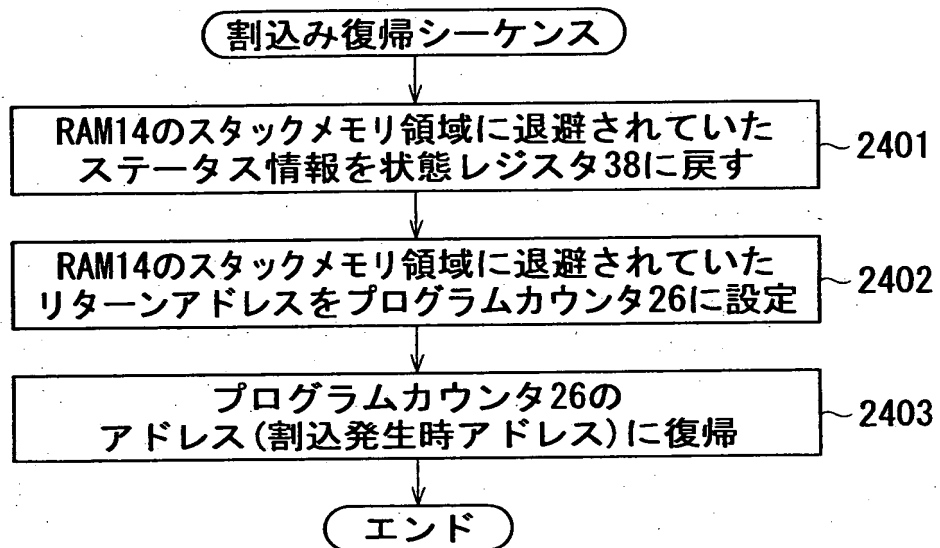
【図 2 2】



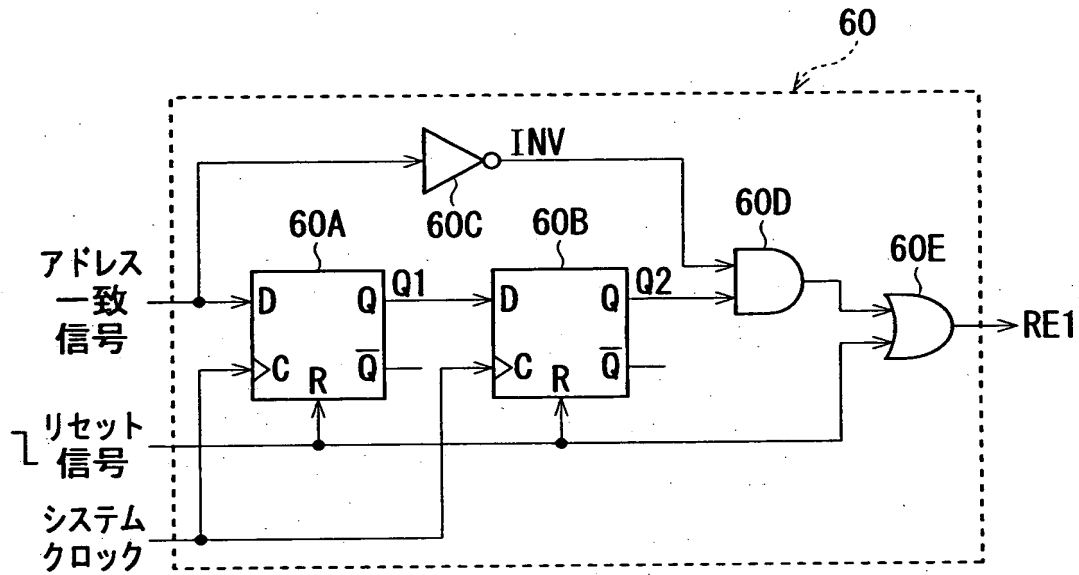
【図 2 3】



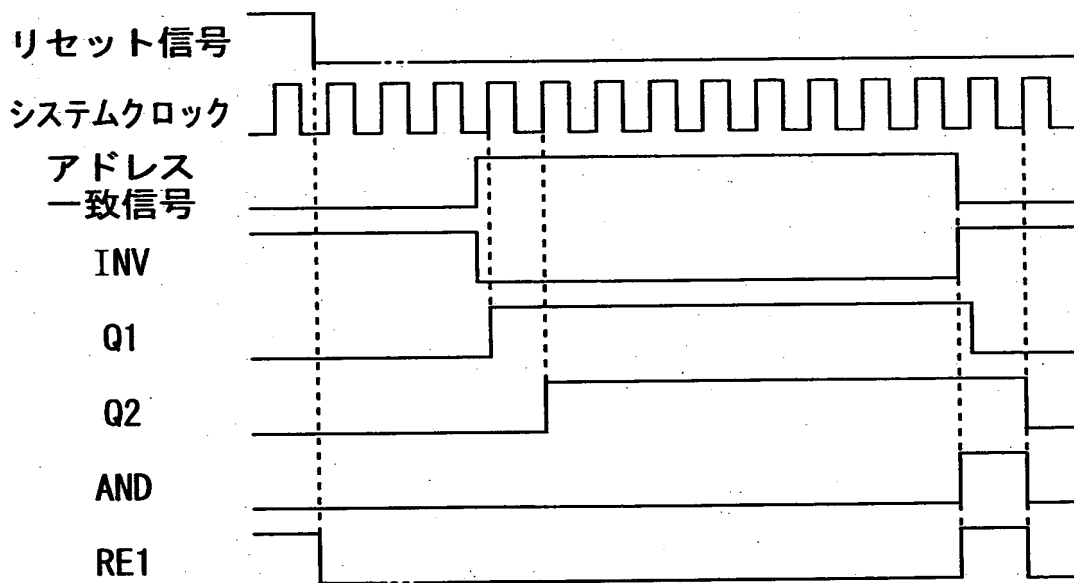
【図 24】



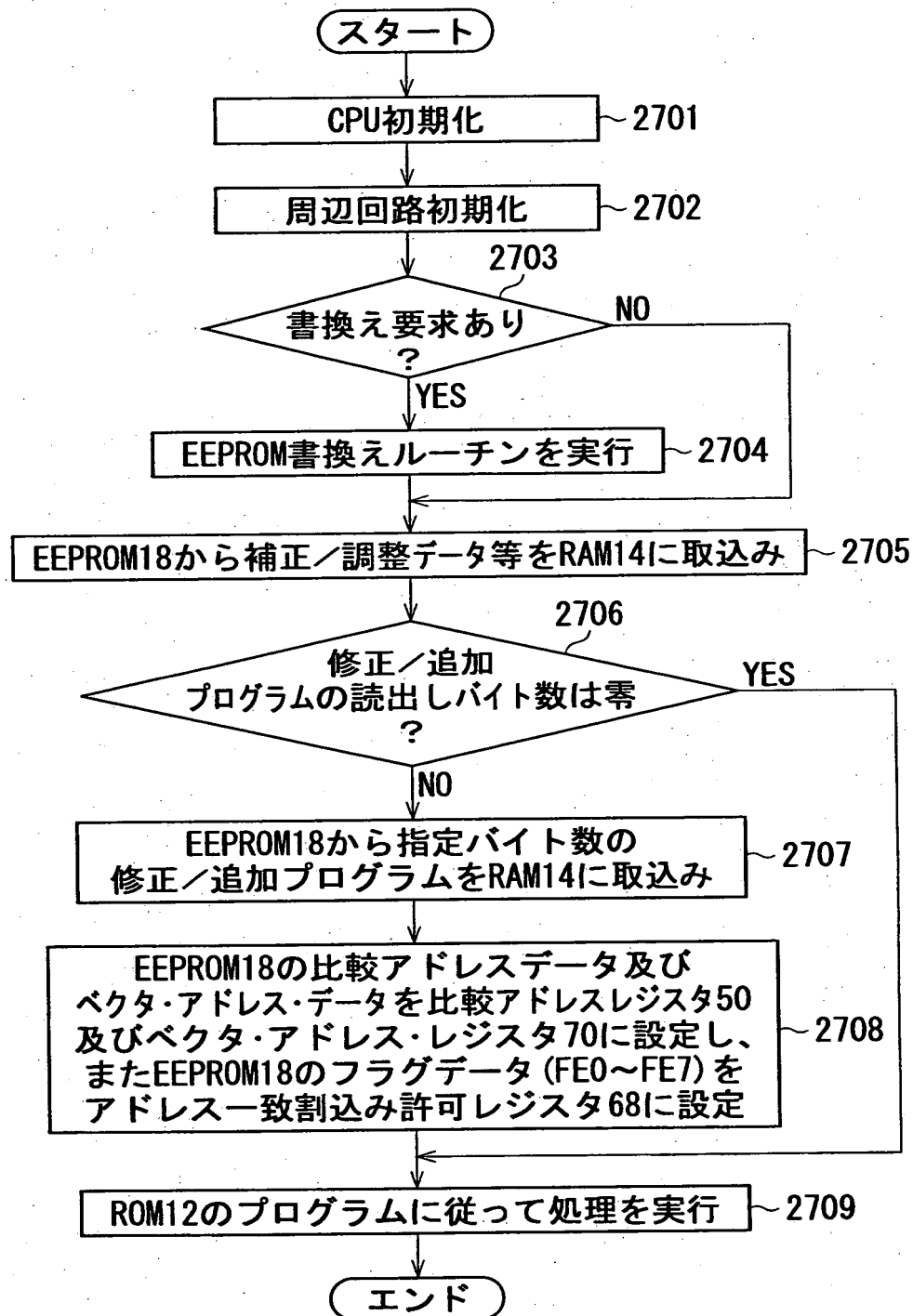
【図 2 5】



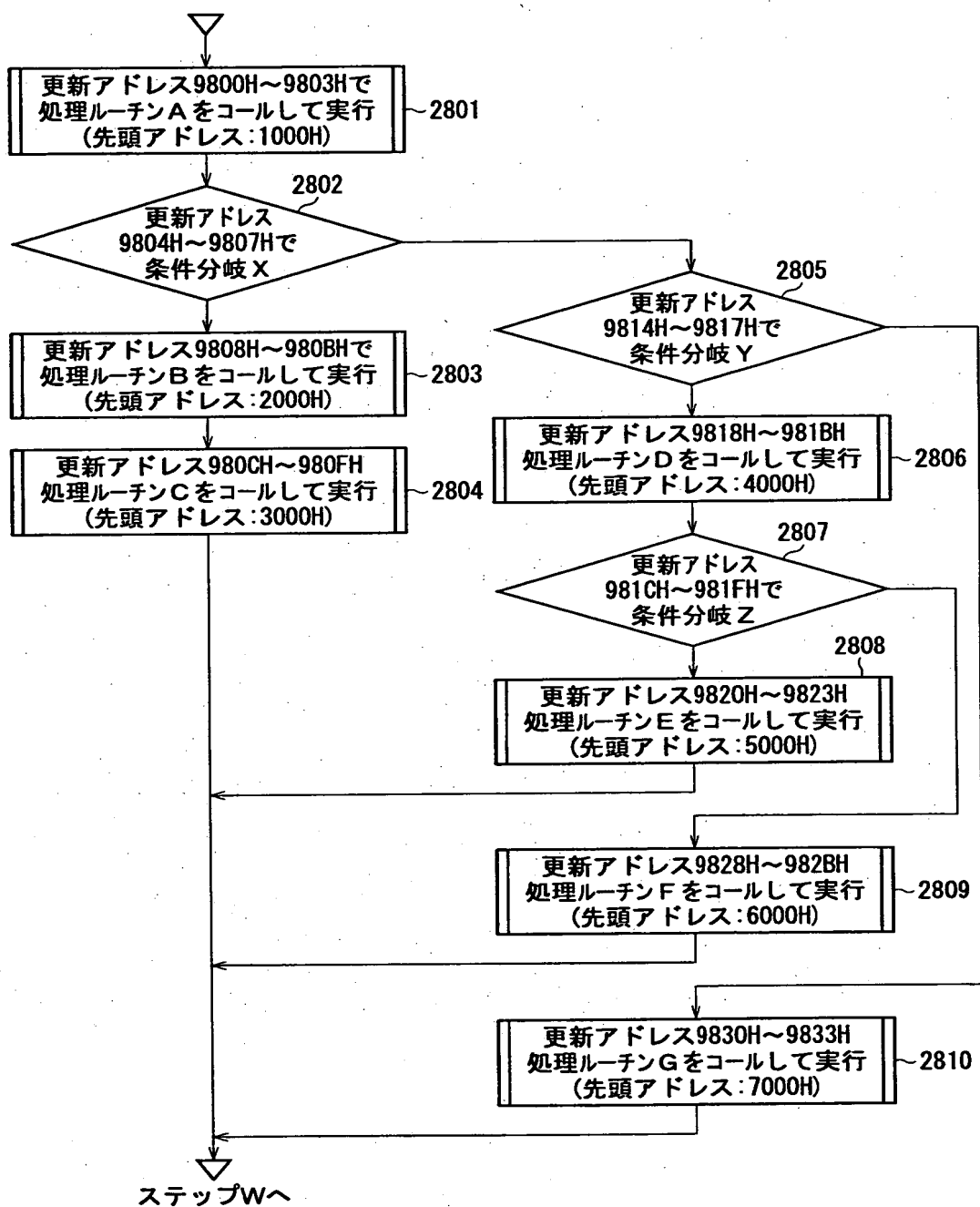
【図 2 6】



【図 27】

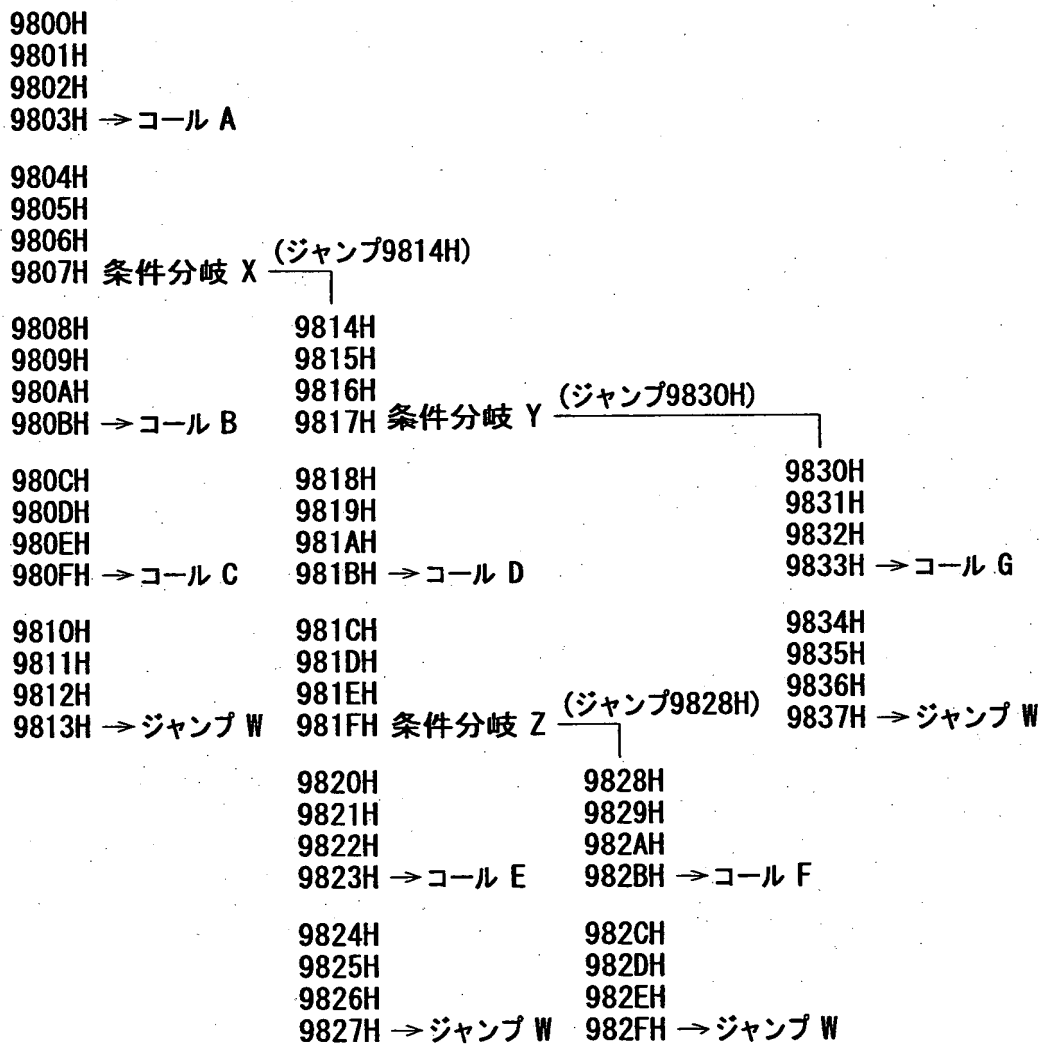


【図 2 8】



【図 2 9】

プログラム
カウンタ 26
更新アドレス

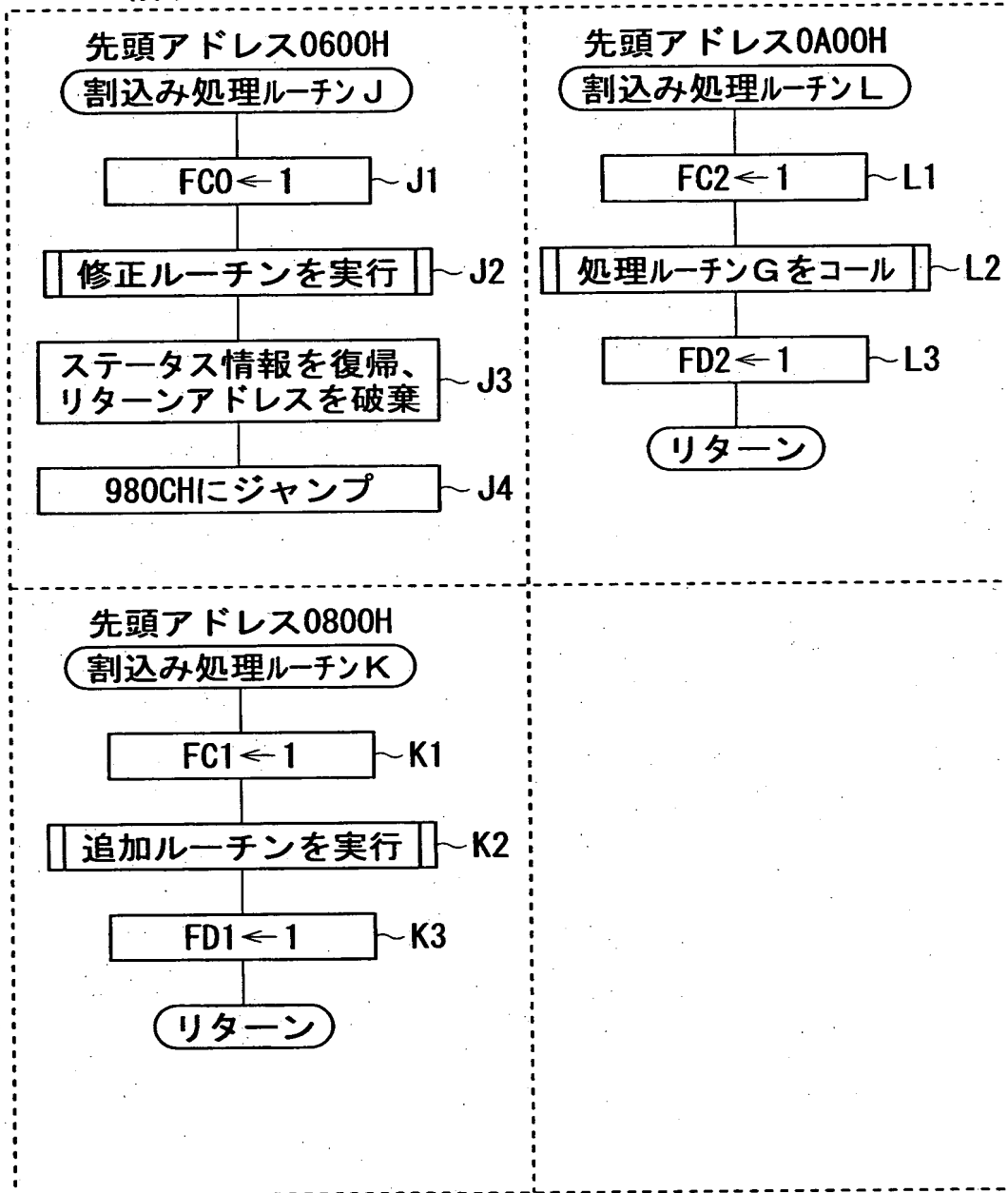


【図 3 0】

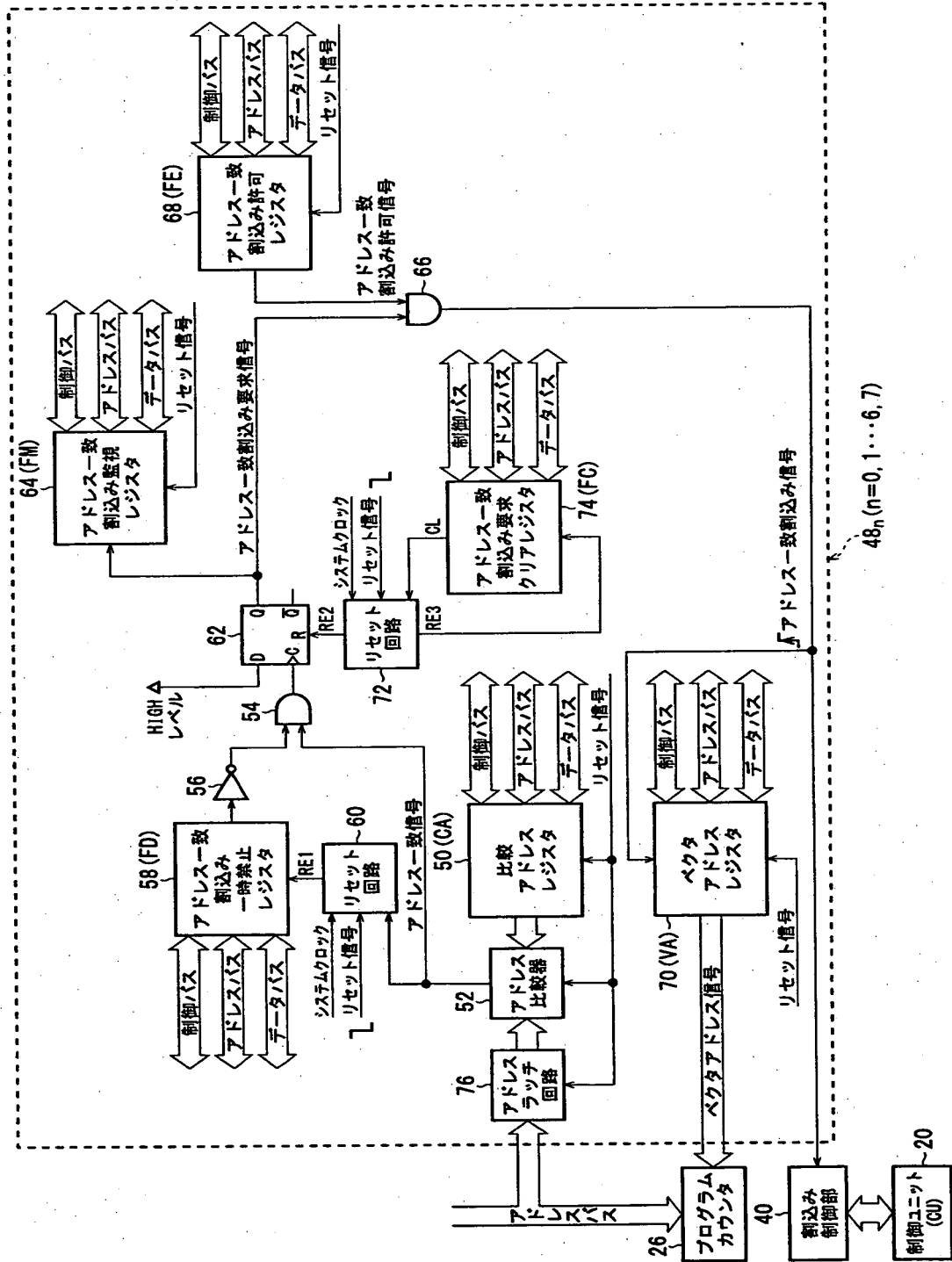


【図 3 1】

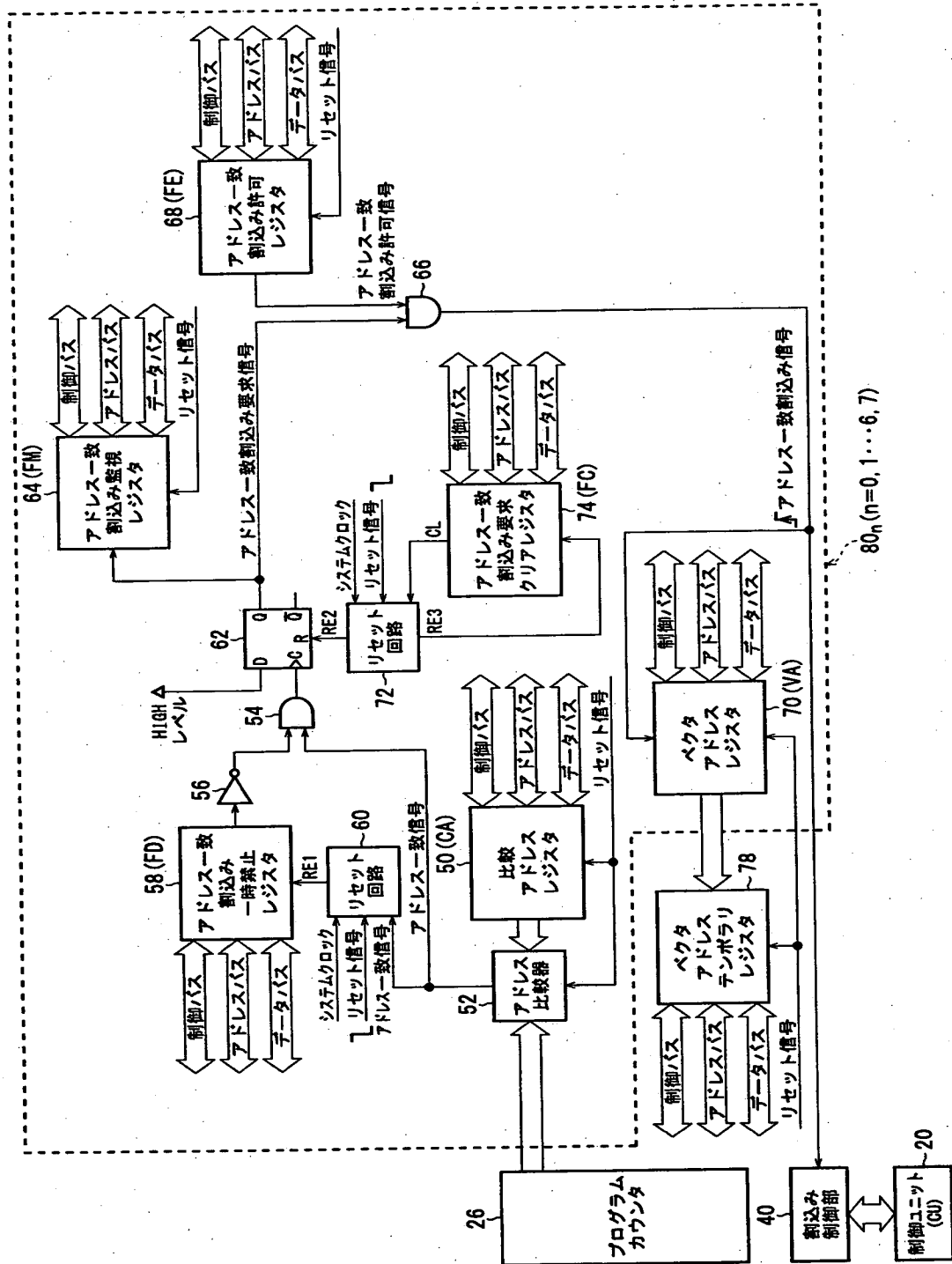
RAM14の領域



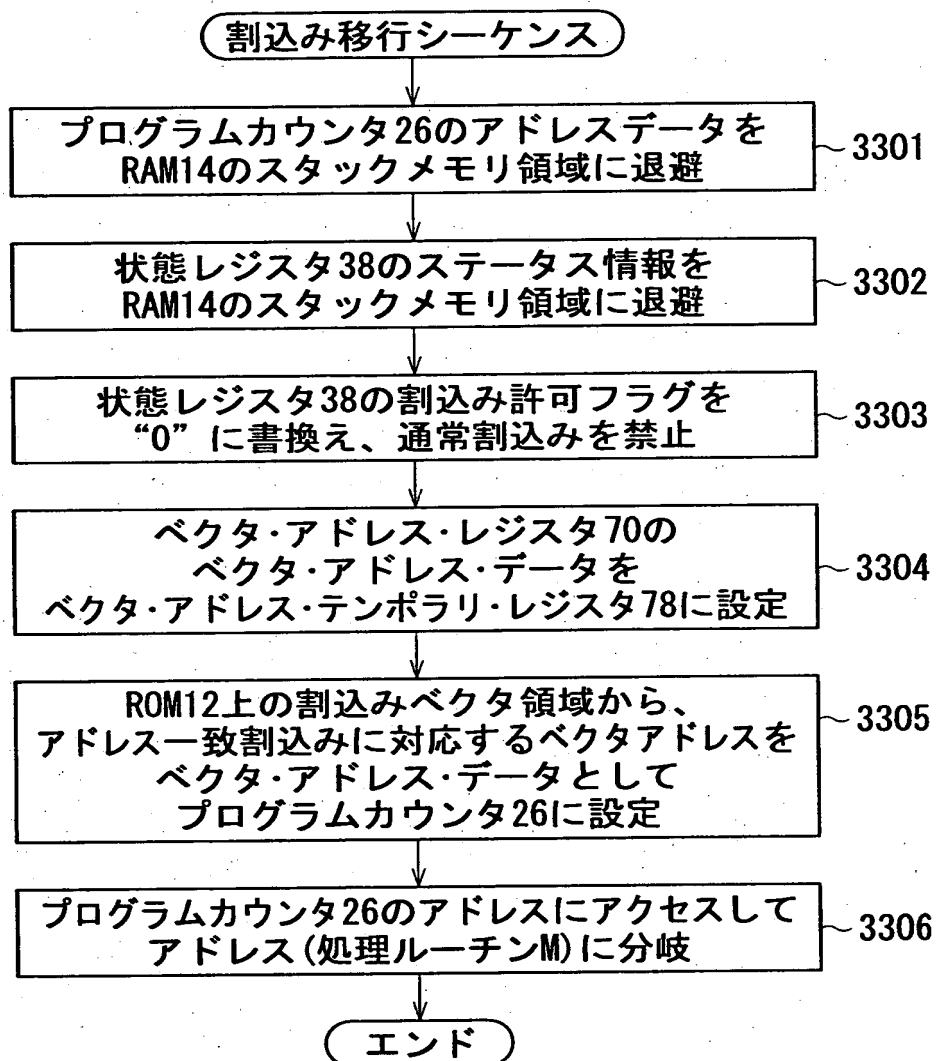
【図 3 2】



【図 3 3】



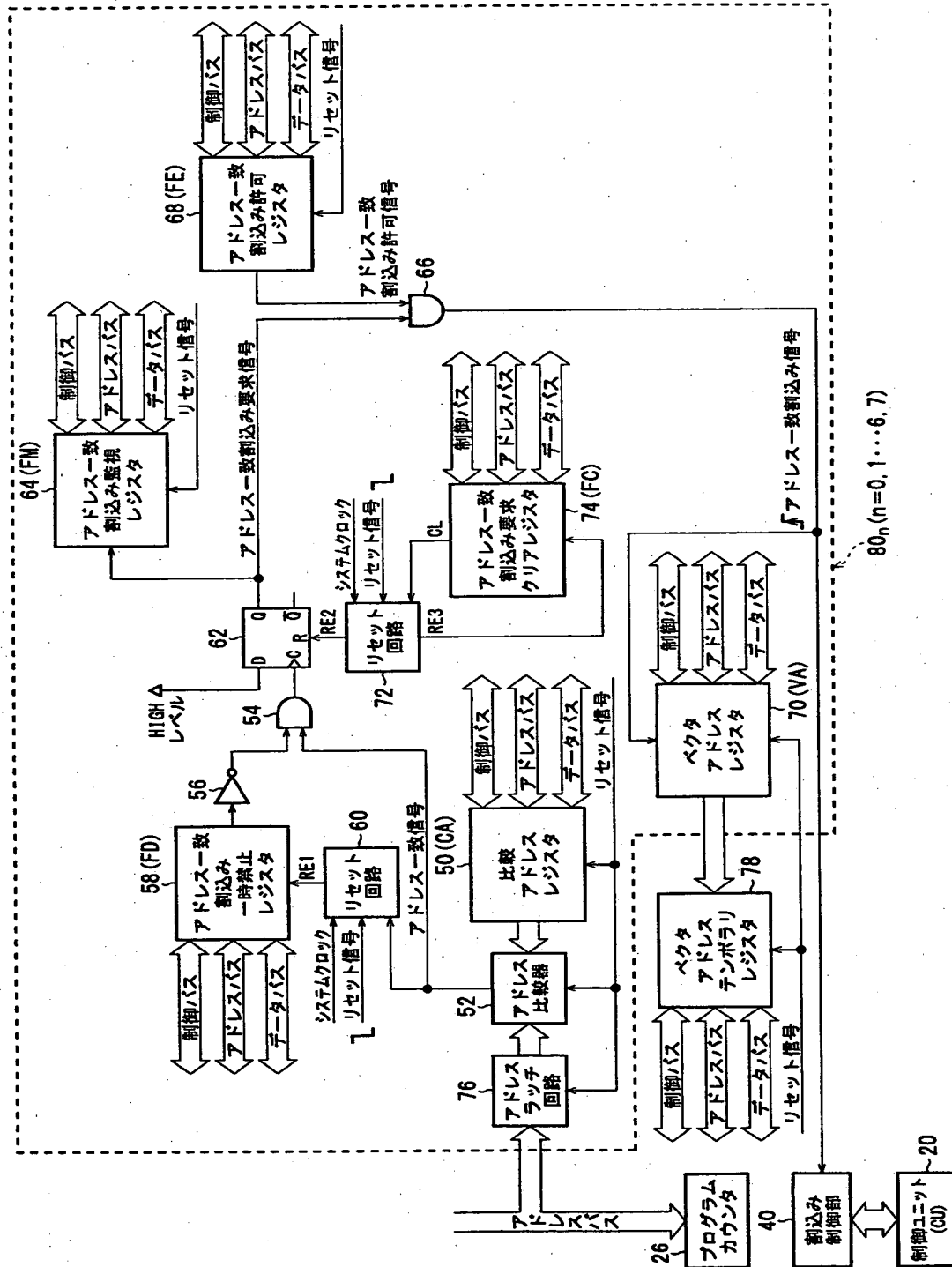
【図 34】



【図 3 5】



【図 3 6】



【書類名】 要約書

【要約】

【課題】 読出し専用メモリ（ROM）のメインプログラムの全ての箇所に対して修正或いは変更を必要に応じて加え得るワン・チップ・マイクロコンピュータを提供する。

【解決手段】 ROM 1 2 にはメインプログラムが格納され、制御／演算手段（2 0、3 2）はメインプログラムの一連のアドレスに逐次アクセスして命令を取り込み、該命令を解読してその解読命令に基づいて逐次処理を行う。レジスタ 5 0 は制御／演算手段によって行われる処理に割込み処理を発生させるべき少なくとも 1 つの割込み発生アドレスを格納し、プログラムカウンタ 2 6 は制御／演算手段によってアクセスすべきアドレスを逐次更新する。プログラムカウンタで逐次更新されるアドレスをレジスタ 5 0 の割込み発生アドレスと比較し、双方のアドレスが一致すると、所定の割込み先アドレスにアクセスする。

【選択図】 図 2

出 願 人 履 歷 情 報

識別番号

[000000527]

1. 変更年月日 1990年 8月10日

[変更理由] 新規登録

住 所 東京都板橋区前野町2丁目36番9号

氏 名 旭光学工業株式会社